Semantic Guidance for Saturation Provers

William McCune[⋆]

Argonne National Laboratory, Illinois, USA
`McCune@mcs.anl.gov`
`http://www.mcs.anl.gov/~mccune/`

Abstract. We use finite interpretations to guide searches in first-order and equational theorem provers. The interpretations are carefully chosen and based on expert knowledge of the problem area of the conjecture. The method has been implemented in the Prover9 system, and equational examples are given the areas of lattice theory, Boolean algebras, and modular ortholattices.

1 Introduction

Automated deduction methods for first-order and equational problems have focused mostly on completeness-preserving restrictions on inference rules, effective term orderings, and special-purpose inference and simplification rules. Less attention has been paid to ordering the search. In saturation systems that use the *given-clause algorithm* or one if its variants, the order of the search is determined mostly by selection of the given clauses. The selection is usually by a weighting function that considers syntactic properties of clauses such as length, depth, occurrences of particular symbols, and patterns of symbols.

We propose to use semantic criteria in addition to syntactic weighting functions to select the given clauses, and we view this method as a form of semantic guidance for theorem provers. The semantic criteria are finite interpretations of the language of the problem.

Interpretations have been used previously for restricting the application of inference rules, for example, a semantic rule may require that one of the parents be false in the interpretation [9, 6, 11]. However, semantic inference rules are frequently incompatible, from both theoretical and practical points of view, with other important methods such as simplification and restrictions based on term orderings.

Semantic guidance with finite interpretations has been used previously, most notably in the recent versions of SCOTT series of provers. In MSCOTT [2], Hodgson and Slaney use semantic guidance with multiple interpretations that are generated automatically and updated during the search. In Son of SCOTT [10], Slaney, Binas, and Price introduced the notion of *soft constraints* that allows the use of just one interpretation that partially models some subset of the

[⋆] Supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-ENG-38.

derived clauses. As in MSCOTT, Son of SCOTT automatically generates the interpretation and updates it during the search. In both systems, the interpretations are small, usually with at most 4 elements.

In this project we are studying the use of larger interpretations that are carefully chosen by the user and fixed throughout the search. The interpretations are intended to be based on expert knowledge on the theory of the problem and on closely related theories. We give several examples of using semantic guidance on difficult equational problems and touch on several ideas for choosing good interpretations.

2 Semantic Strategies and Choice of Interpretations

The roots of semantic strategies for automated theorem proving are in the set of support strategy, introduced by Wos et al in 1965 [14]. The primary motivation for the set of support strategy rests on the assumption that many conjectures have the form *theory, hypotheses ⇒ conclusion.* The idea is that when searching for a proof, one should avoid exploring the theory and focus instead on the hypotheses and the conclusion. The set of support strategy is a semantic restriction strategy, and (assuming that the set of support consists of the hypotheses and conclusion) an arbitrary model of the theory, in which the hypotheses or the denial of the conclusion are false, is used to prove completeness. The effect is that all lines of reasoning must start with the hypotheses or the denial of the conclusion.

Semantic restriction strategies based on other general interpretations (e.g., positive resolution), or on arbitrary explicit interpretations were developed later, most notably by Slagle in 1967 [9]. These rules generally require that one of the parents for each binary inference be false in the interpretation.

The motivation for the present work on semantic guidance with carefully selected interpretations is similar to the motivation for the set of support strategy. If the conjecture has the form *theory, hypotheses ⇒ conclusion*, we wish to focus the search on lines of reasoning that connect the hypotheses to the conclusion. If the theory is true in the guiding interpretation, and the hypotheses and conclusion are false, we believe that lines of reasoning consisting mostly of false clauses will be valuable in connecting the hypotheses to the conclusion.

Because we propose to use semantics to guide rather than restrict the search, valuable consequences of the theory (e.g., lemmas true in the interpretation) can be easily derived, and these may help with the "false" lines of reasoning.

If the conjecture has no obvious hypotheses that are separate from the theory, the interpretation should falsify some part of the theory (a model of the theory that falsifies the conclusion gives a counterexample). In particular, we believe that the interpretation should be a model of a slightly weakened theory, in which the conclusion is false. For example, if the goal is to prove that a theory is associative, one might wish to use a nonassociative interpretation that satisfies many other properties of the theory. If one is unfamiliar with or unsure of the theory, one can use a nonassociative interpretation that satisfies properties of closely related theories.

3 Implementation in Prover9

Prover9 [5] is Otter's [4] successor, and it is similar to Otter in many important ways. In particular, it uses the *given-clause algorithm* (the so-called *Otter loop*), in which weighting functions are used to select the next given clause, that is, the next path to explore. Ordinarily (without semantic guidance), Prover9 cycles through two functions: selecting the oldest clause (to provide a *breadth*-first component) and selecting the lightest clause (to provide a *best*-first component). The ratio of the two functions is determined by parameters.

For semantic guidance, Prover9 accepts one or more finite interpretations in the format produced by Mace4 [3]. Each clause that is retained (input or derived) is evaluated in the interpretations. The clause is marked as *true* if it is true in all of the interpretations; otherwise is it marked as *false*. An exception is that if evaluation of a clause in an interpretation would be too expensive (determined by a parameter that considers the number of variables in the clause and the size of the interpretation), the evaluation is skipped and the clause is marked with the default value *true*. The mark is used when (and only when) selecting given clauses.

When using semantic guidance, Prover9 cycles through three methods when selecting the next given clause: (1) the oldest clause, (2) the lightest *true* clause, and (3) the lightest *false* clause. The ratio of the three methods is determined by parameters. (Son of SCOTT [10] uses a similar 3-way ratio.) We use the notation $A : B : C$ to mean A rounds of selecting the given clause by age, B rounds selecting *true* clauses of lowest weight, and C rounds selecting *false* clauses of lowest weight, and so on. If a *false* clause is called for and none is available, a *true* clause is substituted, and vice versa.

4 Examples

The theorems cited here were first proved with Otter by using various search strategies, with substantial interaction from the user. All of the examples are equational theorems, and a paramodulation inference rule with simplification (demodulation) was used, similar to unfailing Knuth-Bendix completion.

The interpretations for the semantic guidance were produced by Mace4 (quickly) after the user had determined the desired properties for the interpretations. Although the Prover9 implementation can handle multiple interpretations, each of the examples uses just one.

The Prover9 jobs used ratio $1 : 1 : 4$ (*age*:*true*:*false*) for selecting the given clauses. The weighting function for the *true* and *false* components was simply the number of symbols in the clause.

As is frequently done with Prover9, limits were set on the size of equations for several of the searches; these limits substantially improve the performance of

Prover9. The term-ordering method and symbol precedence/weights are usually very important in equational problems, but they are not so important here, because these examples have so few symbols. We used the lexicographic path ordering (LPO) with the default symbol precedence.

Finally, Prover9 was directed to introduce a new constant when it deduced that a constant satisfying some property exists. For example, if $f(x, f(x, x)) = f(y, f(y, y))$ was derived, the equation $f(x, f(x, x)) = c$, for a new constant c was inferred, with c added to the interpretation in such a way that $f(x, f(x, x)) = c$ is true.

Waldmeister [1] is usually assumed to be the fastest automatic prover for equational logic. We ran each example with version 704 (July 2004) of Waldmeister in its automatic mode with a time limit of four hours, and the results are given below with each example. Comparison between provers on a small number of examples is usually not meaningful; the purpose of the Waldmeister jobs is simply to give another measure of the difficulty of these examples.

4.1 Lattice Theory Identities

This example arose in a project to find weak Huntington identities, that is, lattice identities that force a uniquely complemented lattice to be Boolean [8]. The following two identities (among many others) were proved to be Huntington identities, and we then looked at the problem of whether one is weaker than the other.

$$(x \wedge y) \vee (x \wedge z) = x \wedge ((y \wedge (x \vee z)) \vee (z \wedge (x \vee y))) \tag{H82}$$
$$x \wedge (y \vee (x \wedge z)) = x \wedge (y \vee (z \wedge ((x \wedge (y \vee z)) \vee (y \wedge z)))) \tag{H2}$$

Let LT be an equational basis for lattice theory in terms of meet and join. Mace4 easily finds a counterexample to LT, H2 $\Rightarrow$ H82. The statement LT, H82 $\Rightarrow$ H2 is a theorem and is the focus of this example.

This theorem has the form *theory, hypotheses* $\Rightarrow$ *conclusion*, and a natural choice for a guiding interpretation is model of the theory that falsifies the hypothesis. Mace4 easily finds a lattice of size 6 satisfying those constraints, and also shows that there is none smaller and none other of size 6. By using semantic guidance with that lattice, Prover9 proved the theorem in 10 seconds; without semantic guidance, Prover9 proved it in about one hour. Waldmeister proved the theorem in about 5 minutes.

4.2 Boolean Algebra Basis

This example is on Veroff's 2-basis for Boolean algebra in terms of the Sheffer stroke [13].[1] Consider the following equations, where f is the Sheffer stroke.

$$f(x, y) = f(y, x) \tag{C}$$
$$f(f(x, y), f(x, f(y, z))) = x \tag{V}$$
$$f(f(f(y, y), x), f(f(z, z), x)) = f(f(x, f(y, z)), f(x, f(y, z))) \tag{S}$$

[1] The Sheffer stroke, which can be interpreted as the *not-and* or NAND operation, is sufficient to represent Boolean algebra.

The pair C,V is a basis for Boolean algebra, and S is a member of Sheffer's original basis. The theorem for this example is C,V $\Rightarrow$ S. This statement does not have the form *theory, hypotheses* $\Rightarrow$ *conclusion* for any nontrivial and well-understood theory, and it is not so obvious where to look for a guiding interpretation.

When faced with the conjecture, we know that the goal is to prove a property of Boolean algebra. We use an interpretation that is close to, but not, a Boolean algebra. Consider the chain of varieties *ortholattices (OL), orthomodular lattices (OML), modular ortholattices (MOL), Boolean algebras (BA).* (See [7] for Sheffer stroke as well as standard bases for these varieties.) Mace4 can be used to find the smallest MOL that is not a BA. It has size 6, and there is exactly one of size 6. With that interpretation as a guide, Prover9 proved the theorem in about 4 minutes. A similar search without semantic guidance produced a proof in about 6.5 minutes. Waldmeister took about 8 minutes to prove the theorem.

4.3 Modular Ortholattice Single Axiom

The two theorems in this section are on a single axiom for modular ortholattices in terms of the Sheffer stroke [7]. The second theorem is quite difficult; it was first proved by Veroff using the proof sketches method [12], and it was first proved automatically (given a helpful interpretation) with the semantic guidance described here. Consider the following equations, all in terms of the Sheffer stroke.

$$f(f(y,x),f(f(f(x,x),z),f(f(f(f(f(x,y),z),z),x),f(x,u)))) = x \qquad \text{(MOL)}$$
$$f(x,f(f(y,z),f(y,z))) = f(y,f(f(x,z),f(x,z))) \qquad \text{(A)}$$
$$f(x,f(y,f(x,f(z,z)))) = f(x,f(z,f(x,f(y,y)))) \qquad \text{(M)}$$

The equation MOL is a single axiom for modular ortholattices, A is an associativity property, and M is a modularity property. The two theorems in focus are MOL $\Rightarrow$ A and MOL $\Rightarrow$ M. Neither suggests an obvious interpretation for guidance.

In the rest of this section, the term *associative* refers to the operations meet and join when defined in terms of the Sheffer stroke. As in the preceding example, we considered the chain of varieties OL–OML–MOL–BA.

For the first theorem (the associativity property), we chose the smallest nonassociative interpretation that satisfies several MOL (modular ortholattice) properties; it has size 8. With that interpretation as guidance, Prover9 proved the theorem in about 8.5 minutes. Without semantic guidance, Prover9 took about the same amount of time to prove the theorem, but required a much larger search, generating 57% more clauses. Waldmeister did not find a proof of the theorem within the time limit of 4 hours.

For the second theorem (the modularity property), we chose the smallest nonmodular orthomodular lattice, which has size 10. The motivation for this choice is similar to that for the Boolean Algebra 2-basis example, that is, to use an interpretation that is very close to, but not, an algebra in the variety corresponding to the goal of the conjecture. With the interpretation as guidance, Prover9 proved the theorem in about 3.8 hours. Without semantic guidance, Prover9 failed to prove the theorem within 6 hours. Waldmeister proved the theorem in about 3.3 hours.

5 Remarks

The Mace4 jobs that find the interpretations and the Prover9 jobs that find the proofs for the examples can be found on the Web at the following location.

`http://www.mcs.anl.gov/~mccune/papers/semantic-strategy/`

When Prover9 selects a given clause, it is printed to the output file, including any *false* marks. The *false* marks are shown as well in any proofs that are printed. These notations help the user to analyse the effects of semantic guidance.

The output files on the Web show the following.

- The *false* clauses that are selected as given clauses are, for the most part, heavier than the *true* clauses. When the *false* clauses are the same weight as the *true* clauses, the *false* clauses have much higher ID numbers. Both of these properties indicate that the *false* clauses would not be selected so soon without semantic guidance.
- Most of the proofs have a preponderance of *false* clauses, especially near the ends of the proofs. The *true* clauses do, however, play very important roles as lemmas of the theory, suggesting that semantic *restriction* strategies that eliminate many *true* clauses may be less useful than semantic guidance strategies.

Comparisons of the Prover9 jobs with and without semantic guidance indicate that semantic guidance may be more helpful in examples where the interpretations are more obvious or are more carefully chosen. The lattice theory example, which immediately suggests a lattice interpretation falsifying the hypothesis, shows a great improvement. The Boolean Algebra 2-basis and the MOL modularity examples, in which the interpretation is close to the theory corresponding to the goal, show a substantial improvement. The MOL associativity example, which uses a rather ad hoc interpretation which was not carefully chosen, shows only a small improvement.

Future work includes more experimentation with various interpretations, experimentation on non-equational and non-Horn problems, and automating the selection of effective interpretations for semantic guidance.

References

1. T. Hillenbrand, A. Buch, R. Vogt, and B. Löchner. Waldmeister. *J. Automated Reasoning*, 18(2):265–270, 1997.
2. K. Hodgson and J. Slaney. TPTP, CASC and the development of a semantically guided theorem prover. *AI Communications*, 15:135–146, 2002.

3. W. McCune. Mace4 Reference Manual and Guide. Tech. Memo ANL/MCS-TM-264, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, August 2003.
4. W. McCune. Otter 3.3 Reference Manual. Tech. Memo ANL/MCS-TM-263, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, August 2003.
5. W. McCune. Prover9. `http://www.mcs.anl.gov/~mccune/prover9/`, 2005.
6. W. McCune and L. Henschen. Experiments with semantic paramodulation. *J. Automated Reasoning*, 1(3):231–261, 1984.
7. W. McCune, R. Padmanabhan, M. A. Rose, and R. Veroff. Automated discovery of single axioms for ortholattices. *Algebra Universalis*, 52:541–549, 2005.
8. R. Padmanabhan, W. McCune, and R. Veroff. Lattice laws forcing distributivity under unique complementation. *Houston J. Math.* To appear.
9. J. R. Slagle. Automatic theorem proving with renamable and semantic resolution. *J. ACM*, 14(4):687–697, 1967.
10. J. Slaney, A. Binas, and D. Price. Guiding a theorem prover with soft constraints. In *Proceedings of the 16th European Conference on Artificial Intelligence*, pages 221–225, 2004.
11. J. Slaney, E. Lusk, and W. McCune. SCOTT: Semantically constrained Otter. In A. Bundy, editor, *Proceedings of the 12th International Conference on Automated Deduction, Lecture Notes in Artificial Intelligence, Vol. 814*, pages 764–768. Springer-Verlag, 1994.
12. R. Veroff. Solving open questions and other challenge problems using proof sketches. *J. Automated Reasoning*, 27(2):157–174, 2001.
13. R. Veroff. A shortest 2-basis for Boolean algebra in terms of the Sheffer stroke. *J. Automated Reasoning*, 31(1):1–9, 2003.
14. L. Wos, G. Robinson, and D. Carson. Efficiency and completeness of the set of support strategy in theorem proving. *J. ACM*, 12(4):536–541, 1965.