Predicate-Oriented Isomorphism Elimination in Model Finding*

Xiangxue Jia and **Jian Zhang**
Laboratory of Computer Science
Institute of Software
Chinese Academy of Sciences
Beijing 100080, China
Email: {jxx,zj}@ios.ac.cn

Abstract

Finding models of logical formulas is a challenging problem. For first-order formulas, a finite model can be found by exhaustive search. For many structured problem instances, there is much isomorphism in the search space. This paper proposes general-purpose techniques for eliminating isomorphic subspaces, which can be helpful when the formulas have many predicates. The techniques are based on inherent symmetries in first-order clauses.

1 Introduction

Satisfiability checking (or model finding) is an important problem in logic and Artificial Intelligence. If restricted to the propositional logic, the problem is well-known as SAT, and has been studied by many people. In this paper, we focus on first-order satisfiability problems. More specifically, we study how to find finite models of first-order formulas. We propose two methods which use symmetries to eliminate isomorphic subspaces and prune the search tree. One technique works row-by-row; the other one processes submatrices, based on Ramsey numbers.

Symmetry breaking can be very helpful to improve the efficiency of combinatorial searching. For example, it has played a key role in the solution of some previously open questions on quasigroups [Fujita *et al.*, 1993]. But the technique used there is problem-specific. Other researchers have studied more general methods for using symmetries to improve the efficiency of model finding. The Least Number Heuristic (LNH) [Zhang and Zhang, 1995] is such a technique. It works well on many problems (including the quasigroup problems). But it is not effective on predicates. The techniques proposed in this paper can remedy this weakness to some extent.

2 Background

We assume that the reader is familiar with the basic concepts of classical logics (including propositional and first-order logics). As mentioned earlier, this paper deals with the problem of finding finite models of first-order formulas. Currently, there are some automated tools available for doing this. Tools like SEM [Zhang and Zhang, 1995] search for the models directly, while other tools (like MACE2 [McCune, 2001]) translate the problem into SAT and use efficient SAT solvers.

For simplicity, we assume that there is only one domain: $D_n = \{\ 0,\ 1,\ \ldots,\ n-1\ \}$. (But the methods can also be used on many-sorted formulas.) Suppose there is a binary predicate, P, in the formulas. To give an interpretation for P, we need to find an $n \times n$ matrix, in which each entry (also called a *cell*) has a truth value (TRUE or FALSE).

A set of ground clauses, Φ, is *symmetric* with respect to a set of domain elements, if Φ remains the same under any permutation of the elements. When Φ is obvious from the context, we also say that the elements are *symmetric* or *interchangeable* with each other. Clearly, this is an equivalence relation defined on a set of domain elements.

For convenience, we assume that all the domain elements are symmetric initially. We may think that there is only one equivalence class on the domain elements. This class includes every element of D_n.

Ramsey Numbers

Ramsey numbers are well-known in graph theory. For a pair of positive integers p and q ($p > 1, q > 1$), the Ramsey number $r = R(p, q)$ is the minimum number of vertices such that all undirected simple graphs of order r contain a clique of order p or an independent set of order q.

Currently we do not know the exact values of many Ramsey numbers. But we know that

$$R(2,q) = q \qquad \text{for any } q,$$
$$R(3,3) = 6, \quad R(3,4) = 9, \quad R(3,5) = 14, \ldots$$
$$R(4,4) = 18, \quad R(4,5) = 25$$

and obviously, for any p and q, $R(p, q) = R(q, p)$.

3 Row-by-row Isomorphism Elimination

Suppose there is a model. Let us consider an interpretation for P. First look at the first row of the matrix, that is, the value of $P(0, y)$ (for $y > 0$). For these $(n - 1)$ truth values, either of the following propositions must hold, but not both:

(1) At least $\lceil (n-1)/2 \rceil$ cells got the value TRUE;

(2) At least $\lceil n/2 \rceil$ cells got the value FALSE.

Here $\lfloor x \rfloor$ denotes the largest integer i such that $i \le x$, and $\lceil y \rceil$ denotes the smallest integer j such that $j \ge y$.

*Supported by the NSF of China (No. 60125207).

If (1) is true, without loss of generality, we can assume that the cells

$$P(0, \lfloor (n-1)/2 \rfloor + 1), \quad \ldots, \quad P(0, n-1)$$

take the value TRUE. Similarly, if (2) is true, we assume that the second half of the first row got the value FALSE. So we have two subcases.

The above kind of "divide-and-conquer" can be repeated until the size of the equivalence class is reduced to 1. And we can get many subcases.

We call this technique the *Row-by-Row (RR) strategy*. Now we prove its correctness. Let GC denote the set of ground clauses obtained by instantiating the first-order clauses using all the elements in the domain D_n.

Lemma 1. Suppose $1 < t \leq n-1$, $0 \leq k < n-t$, and GC is symmetric with respect to $\{ n-t, \ldots, n-1 \}$. Then any model can be mapped (under certain permutation of domain elements) to one of the following two models:

(1) a model in which $P(k, n-r_1), \ldots, P(k, n-1)$ are TRUE, where $r_1 = \lceil t/2 \rceil$;

(2) a model in which $P(k, n-r_2), \ldots, P(k, n-1)$ are FALSE, where $r_2 = \lceil (t+1)/2 \rceil$.

Theorem 1. If GC is symmetric with respect to D_n, there is a model of GC in D_n if and only if a model exists in one of the subcases.

For a given row, there can be several ways of dividing the assignments into two subcases. For example, we could divide the assignments at the first row into these two subcases:

(1) $P(0, 1)$ is TRUE;

(2) $P(0, y)$ is FALSE, for every $y \geq 1$.

4 A Technique Based on Ramsey Numbers

Now we present a search strategy based on the Ramsey numbers, which is called the *RN strategy*. It tries to determine the values of several submatrices, starting from the upper-left corner to the lower-right corner.

If P is commutative, i.e., $P(x, y) = P(y, x)$ for any x, y, then the matrix can be regarded as a complete graph whose edges are colored with TRUE or FALSE. When n is greater than or equal to the Ramsey number $R(p, q)$, there will be a complete subgraph K_p with color TRUE, or a complete subgraph K_q with color FALSE. Without loss of generality, we can divide the model finding process into two subcases:

(1) For any i, j $(0 \leq i, j < p, i \neq j)$, $P(i, j)$ and $P(j, i)$ are TRUE.

(2) For any i, j $(0 \leq i, j < q, i \neq j)$, $P(i, j)$ and $P(j, i)$ are FALSE.

When n is a bigger number, we can apply this kind of "divide-and-conquer" several times.

If P is not commutative, then $P(x, y)$ $(x < y)$ can be regarded as a complete undirected graph whose edges are colored with TRUE or FALSE. Similarly, we can also divide the model searching process into two subcases:

(1) For any i, j $(0 \leq i < j < p)$, $P(i, j)$ is TRUE.

(2) For any i, j $(0 \leq i < j < q)$, $P(i, j)$ is FALSE.

To avoid repeated search, we can put the following condition on the first subcase: There is no K_q colored with FALSE.

5 Experiments

We have tested our techniques on several problems, including

- Pigeon-hole
- Ramsey Number Searching
- Steiner Triple System

Several programs are used, e.g., SEM [Zhang and Zhang, 1995], MACE2 [McCune, 2001] and the SAT solver siege_v4 (`http://www.cs.sfu.ca/~loryan/personal/`).

Experimental results show that, typically our techniques can reduce the search time by a factor of 2 to 8.

6 Concluding Remarks

For many structured problems, if they are specified naturally using first-order logic, the specification suggests a lot of symmetries which can be used to prune the search space. This paper proposes two techniques for using such symmetries. They are mainly aimed at predicates. One of them works row by row, and the other one processes submatrices from the upper-left corner to the lower-right corner. The two techniques may be regarded as complementary to the least number heuristic and its extension. They can be helpful to both first-order and propositional solvers.

Symmetries have also been studied in the SAT community. For example, Aloul *et al.* [Aloul *et al.*, 2003] have obtained very impressive results on various problems in the EDA field. They have implemented a software package called Shatter which can add certain clauses to a SAT instance to make it easier. Such an approach typically has to search for symmetries in large CNF formulas. For some SAT instances, the search may take a long time. Compared with a propositional formulation, a specification in the first-order logic allows one to use symmetries without searching for them.

In the future, we are going to study how to combine various techniques in a single framework. It is also interesting to compare first-order techniques with symmetry-breaking predicates used for SAT solvers.

References

[Aloul *et al.*, 2003] Fadi A. Aloul, Arathi Ramani, Igor L. Markov, and Karem A. Sakallah. Solving difficult instances of boolean satisfiability in the presence of symmetry. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 22(9):1117–1137, 2003.

[Fujita *et al.*, 1993] Masayuki Fujita, John Slaney, and Frank Bennett. Automatic generation of some results in finite algebra. In *Proceedings of the 13th Int'l Joint Conf. on Artif. Intel. (IJCAI)*, pages 52–57. Morgan Kaufmann, 1993.

[McCune, 2001] William McCune. *MACE 2.0 Reference Manual and Guide*. Technical Memo. ANL/MCS-TM-249, Argonne National Laboratory, Argonne, IL, 2001.

[Zhang and Zhang, 1995] Jian Zhang and Hantao Zhang. SEM: a system for enumerating models. In *Proceedings of the 14th Int'l Joint Conf. on Artif. Intel. (IJCAI)*, pages 298–303. Morgan Kaufmann, 1995.