Symmetry in finite models of first order logic

Gilles Audemard and Belaid Benhamou

Centre de Mathematiques et d'Informatique
39, Rue Joliot Curie - 13453 Marseille cedex 13 - France
Tel : 04 91 11 36 25 - Fax : 04 91 11 36 02
{audemard, benhamou}@lim.univ-mrs.fr

Abstract. Systems like Falcon, SEM and FMSET use the known LNH (Least Number Heuristic) heuristic to eliminate some trivial symmetries. We study in this paper a more general notion of symmetry for finite model search in first order logic. We give an efficient detection method for the more general symmetry and show its combination with the trivial symmetry. This increases the efficiency of finite model search generators. The method SEM with and without symmetry is experimented on several interesting mathematic problems to show the advantage of reasoning by symmetry.

1 Introduction

It is well known that model generation of first order logic theories is a difficult problem, undecidable in general. In this paper, we restrict our study to finite model generation.

A finite model of a first order theory is an interpretation of the variables, the functional symbols and the predicates over a finite domain of individuals that satisfies the theory. A finite model generator is an automated tool computing such interpretations. Model generation can be used to prove the consistency of a first order theory (i.e., a model exists) or to find a counter model to disprove its validity. It can find a counter example of some expected conjecture and can help theorem provers to refute validity of formulas. In this sense, finite model generation is complementary to theorem proving.

Symmetry elimination is crucial in finite model generation. To get efficient model generators one has to detect and exploit symmetrical structures. Different approaches to symmetry elimination are used: For instance, the static approach used by James Crawford and al. in [5] for propositional logic theories consists of adding constraints expressing symmetry. The same technique is used by Masayuki Fujita and al. in [6] to search for finite models of quasi-group problems. One drawback of this approach is that only geometric symmetries of the problem are considered.

Many other symmetries appear during the search phase and dynamic approaches use them to reduce the search space. For example the FMC method [7] eliminates some symmetries during the search to avoid isomorphic interpretations. Finite model generators like SEM [11] and FMSET[2] use the LNH (Least

Number Heuristic) heuristic [11] to suppress some trivial symmetries existing between individuals of the domains. The methods use the CNF form of first order logic where all the variables are universally quantified to express problems. As a consequence of this quantification, the individuals in the domains are all trivially symmetrical. That is, if during the search all the first individuals $\{0, \ldots, mdn\}$ of an ordered finite domain $D_n = \{0, \ldots, n-1\}$ (with $0 \leq mdn \leq n-1$) are used in the partial model, then only the individuals of the part $\{mdn+1, \ldots, n-1\}$ remain symmetrical —all the individuals are symmetrical before starting the model search ($mdn = 0$). To keep a maximum number of symmetrical individuals, the LNH heuristic assign in priority cells whose individuals have already been used.

Such trivial symmetries are very useful but they disappear as soon as the first cells are instantiated. The propagation process forces new individuals to be used, which increases the mdn and decreases the subset $\{mdn+1, \ldots, n-1\}$ of symmetrical individuals. This subset becomes empty when mdn reaches the value $n-1$. On the other hand, the subset $\{0, \ldots, mdn\}$ of used individuals increases and become quickly identical to the whole domain D_n.

A lot of other symmetries exist between individuals of the part $\{0, \ldots, mdn\}$ which can be exploited to increase the efficiency of model generation. We show in this work how to detect and use such symmetries, and how to combine them with the trivial ones to maximize the symmetry gain.

This article is organized as follows: In section 2, we give a short description of many-sorted first order logic theories. In section 3, we study the principle of symmetry for finite model and prove that the trivial symmetry is a particular case of our study. We describe the exploitation of this symmetry in section 4. Section 5 presents an experimental evaluation of the known finite model generator SEM method with and without the advantage of symmetry on mathematical problems.

2 Many-sorted first order logic theories

As a knowledge representation, we use the many-sorted first order logic with equality in clausal normal form (CNF) where all the variables are universally quantified. Formally, a many-sorted theory is a triple $T = (S, F, C)$ where S is the set of sorts, F is the set of function and predecate symbols, and C is the set of clauses. We denote by C^t the set of ground instances of C. Without loss of generality, all elements of a sort $s \in S$ of cardinality n are represented by the set $\{0, \ldots, n-1\}$. A terminal term (or a cell) is a term with the following form: $f(e_1, \ldots, e_k)$ where each argument e_i is in some $s_i \in S$ and $f \in F$. The equation $f(e_1, \ldots, e_k) = e$ represents the instantiation of the terminal term $f(e_1, \ldots, e_k)$ to the value e.

A finite interpretation of the theory T is an instantiation of the cells over a domain D_n which can be represented by the operation table of the different function symbols. An interpretation over a finite domain D_n which satisfies all the clauses of C^t is a finite model. If C^t is a terminal set of clauses and I the current partial interpretation then C^t_I denotes the set of clauses C^t simplified by I and T_I the resulting theory.

Example 1. Let be a theory T= $(\{D_n\}, \{h\}, C)$ where $D_n = \{0...n-1\}$ and where C contains the two following axioms.

- $\forall x, \quad h(x, x) = x$
- $\forall x, \forall y \quad h(h(x, y), x) = y$

When $n = 4$ this theory admits a lot of models, among which the one given by the functional table of h denoted I.

h	0	1	2	3
0	0	2	3	1
1	3	1	0	2
2	1	3	2	0
3	2	0	1	3

This is interpreted as the instantiations: $h(0, 0) = 0$, $h(0, 1) = 2$, . . . etc.

3 Permutations and Symmetries

First, we give some basic notions on permutations and define the symmetry principle for individuals of the domains.

Definition 1. *Given a finite set E, a permutation σ of E is a bijection mapping from E onto E. We note $Perm(E)$ the set of all permutations of E.*

Definition 2. *A permutation σ on the set of sorts is defined as $\sigma = (\sigma_1, \ldots, \sigma_n)$ where each $\sigma_i \in Perm(s_i)$*

We now give the definition of symmetry.

Definition 3. *Let $T = (S, F, C)$ be a theory, I the current instantiation and σ a permutation defined on the set of sorts S. The permutation σ is a symmetry of T_I if the following two conditions hold:*

1. *$\sigma(I) = I$, and*
2. *$\sigma(C^t_I) = C^t_I$*

Definition 3 gives new conditions of symmetry in comparison to the propositional case [3]. To verify the symmetry conditions at a given node of the search tree, the permutation has to leave invariant both the current set of terminal clauses and the current instantiation.

Definition 4. *Let $T = (S, F, C)$ be a theory. Two individuals e_1 and e_2 of a sort s are symmetrical if there exists a symmetry σ of the theory such that $\sigma(e_1) = e_2$.*

Definition 5. *Let $T = (S, F, C)$ be a theory. Two interpretations I and J are symmetrical if there exists a symmetry σ such that $\sigma(I) = J$.*

If I and J are symmetrical interpretations ($\sigma(I) = J$) of the theory T then I is a model of T if and only J is a model of the theory T too.

3.1 Symmetry detection

The symmetry detection algorithm consists in two steps: The first one partitions the individuals of the different sorts into primary classes with respect to the necessary conditions of Proposition 1. Two individuals will be candidates to symmetry if they are in the same primary class. The second step is a backtrack search which build the symmetry.

To be symmetrical, two individuals of a same domain have to satisfy some necessary conditions (Proposition 1).

Definition 6. *If I is a partial interpretation of the theory $T = (S, F, C)$, e an individual and $f \in F$ a function, then $\#_{I_{f_i}}(e)$ denotes the number of occurrences of e in I as the ith position argument of the function f and $\#_{I_{f_{val}}}(e)$ denotes the number of occurrences of e in I as a value of the function f.*

Example 2. In example 1, we have $\#_{I_{h_1}}(0) = 4$, $\#_{I_{h_2}}(0) = 4$, and $\#_{I_{h_{val}}}(0) = 4$. In this interpretation we have $\#_{I_{h_i}}(a) = 4$, $\forall i \in \{1, 2\}$, $\forall a \in \{0, 1, 2, 3\}$ and $\#_{I_{h_{val}}}(a) = 4$, $\forall a \in \{0, 1, 2, 3\}$.

Proposition 1 (Necessary Conditions). *Let $T = (S, F, C)$ be a theory and $a \in s$ and $b \in s$ two individuals of the sort $s \in S$. If I is the current interpretation of T, then the individuals a and b are symmetrical if the following condition hold.*

- $\forall f \in I \cap F, \quad \forall i \in \{1, \ldots, arity(f)\}, \quad \#_{I_{f_i}}(a) = \#_{I_{f_i}}(b)$
- $\forall f \in I \cap F, \quad \#_{I_{f_{val}}}(a) = \#_{I_{f_{val}}}(b)$.

Verification of the necessary conditions is an important step of the symmetry detection algorithm. They allow to reduce drastically the permutation search space by partitioning the domains to equivalence classes of individuals which are potential candidates for symmetry.

In the second step, we compute the permutation (the symmetry) using the equivalent classes defined previously. A symmetry in propositional calculus [3] is a permutation which keeps the set of clauses invariant. Such condition can be time consuming when the number of clauses grows. This condition is simplified in finite model generation. Indeed, if the set of clauses contains no individual constant, it is sufficient to look for permutations which leave invariant the partial model. The invariance of the current set of clauses is guaranteed. This avoids checking its invariance. Formally:

Proposition 2. *Let $T = (S, F, C)$ be a theory, I the current interpretation and $\sigma \in Perm(S)$. If C contains no individual of any type $s \in S$ and if $\sigma(I) = I$ then $\sigma(C_I^t) = C_I^t$.*

Proposition 2 gives an important result for symmetry detection, which simplifies the sufficient condition of Definition 3. This increases drastically the efficiency of symmetry detection.

We show now that the trivial symmetry treated by the LNH heuristic is a particular case of our symmetry notion.

Proposition 3. *If $T = (S, F, C)$ is a theory, I the current instantiation and σ a trivial symmetry, then $\sigma(I) = I$.*

Proof. The trivial symmetry σ is formed by the cycle $(mdn + 1, \ldots, n - 1)$ of the not used individuals. The permutation σ does not permute the individuals $\{0, \ldots, mdn\}$ which are the only ones used in I. I is independent on the permutation σ, thus $\sigma(I) = I$. □

By the same way, we can show that the symmetry treated by the extension XLNH [1] of LNH is also a particular case of our study.

4 Symmetry exploitation

We now give some properties which allow to use the detected symmetries to improve finite model search efficiency.

Proposition 4. *Let I be the current instantiation, a permutation $\sigma \in Perm(S)$ such that $\sigma(I) = I$ and a non instantiated cell ce. If the current instantiation is $ce = e$ then both partial models $I \cup \{ce = e\}$ and $I \cup \{\sigma(ce) = \sigma(e)\}$ are symmetrical.*

Therefore, if we are looking for non symmetrical models then we keep only one of the previous extensions. As we are also interesting on coherence of theories, we give in the following proposition the relationship between the previous extensions.

Proposition 5. *If the partial interpretation I of the theory $T = (S, F, C)$ is coherent, then [the extension $I \cup \{ce = e\}$ is coherent if and only if $I \cup \{\sigma(ce = e)\}$ is coherent]*

Proposition 5 gives an important result that we exploit to reduce the search space of a finite model generation. If the instantiation $ce = e$ is contradictory then the instantiation $\sigma(ce = e)$ is contradictory too. We avoid then exploring the branch corresponding to the assignment $\sigma(ce) = \sigma(e)$. That is what we call the symmetry cut. A symmetry of order k can make $k - 1$ cuts in a search tree. To be efficient, one has to detect symmetries with great orders.

Combination of the detected symmetry (DSYM) with the trivial one

Consider a theory with one sort whose individuals are in $D_n = \{0, \ldots, n-1\}$ and I the current interpretation. If the individuals used at the current node of the search tree are all in the part $\{0, \ldots, mdn\}$, $(0 \leq mdn \leq n - 1)$, then all the extensions $I \cup \{t = mdn + 1\}, \ldots, I \cup \{t = n - 1\}$ are trivially symmetrical. These trivial symmetries are exploited by the LNH heuristic. We add to these, the symmetries that we detect on the part $\{0, \ldots, mdn\}$ of used individuals. The trivial symmetry and the one we detect here are independent, since they are defined on two disjoint parts of the domain. Their combination is then trivial and $n - mdn - 1 + k$ symmetry cuts can be made when they are associated.

5 Experimentations

We have experimented the finite model generator SEM [11] with and without the advantage of the detected symmetry and we obtained promizing results.

We first describe in section 5.1 the problems we experimented and then show how symmetry increases the performances of SEM. To know the number of models suppressed by symmetry, our implementation enumerates the total set of models. All CPU times are given in seconds and are obtained on a K6II (400MHz, 128Mb) running under Linux 2.2. The code is written in C. The symbol '-' means that the problem is not solved in less than 2 hours.

5.1 Problems

We used mathematical problems partially described by Jian Zhang in [10] and [9]. All this problems have only one type. The symbol AG is for Abelian group, NG for non Abelian group, GRP a non Abelian group which satisfied $(xy)^4 = x^4y^4$. RU is for unit ring, RNB for non boolean ring which satisfied $x^7 = x$ and RNA a non boolean ring with a counter example of associativity. We have listed axioms of problems AG and RU in table 1. Other problems are generated by the addition of different axioms.

Table 1. Problems AG and RU

Abelian Group	Unit Ring
	$s(x, 0) = x$
$h(x, 0) = x$	$s(x, y) = s(y, x)$
$h(0, x) = x$	$s(x, m(x)) = 0$
$h(x, g(x)) = 0$	$s(x, s(y, z)) = s(s(x, y), z)$
$h(g(x), x) = 0$	$p(x, 1) = x$
$h(h(x, y), z) = h(x, h(y, z))$	$p(x, y) = p(y, x)$
$h(x, y) = h(y, x)$	$s(p(x, y), p(x, z)) = p(x, s(y, z))$
	$s(p(x, z), p(y, z)) = p(s(x, y), z)$

5.2 SEM with and without the advantage of the detected symmetries

Tables 2 and 3 show the experimental results obtained with our first implementation. We can see that SEM with symmetry outperforms SEM without symmetry on the problems NAG, GRP, RU and RNB. With the advantage of the detected symmetry, we generate models of a great size that SEM can not find in a reasonable time.

The experiments confirm that there exist a lot of other symmetries than the trivial ones that LNH does not eliminate. Combining both kinds of symmetry seems to be a promising approach to solve higher size problems.

Table 2. Groups - Comparison.

		SEM with symmetry			SEM without symmetry		
Pb	size	Nb Models	Time	Nodes	Nb Models	Time	Nodes
AG	28	35	246	148 096	162	328	642 103
	30	64	368	189 554	262	559	870 097
	32	551	1 598	351 800	2 295	940	2 037 525
NAG	22	1	211	339 105	3	413	849 128
	24	357	1 146	1 251 884	1 130	2 636	3 526 828
	26	2	1 433	1 432 737	3	3 260	4 252 561
	28	14	2 465	2 065 498	51	6 934	8359103
	30	51	4 127	2 847 643	-	-	-
	31	0	4 737	2 876 674	-	-	-
GRP	24	26	651	647 606	48	2 018	2 419 623
	26	0	727	674 359	0	2 322	2 536 477
	28	0	917	712 491	0	2 794	2 635 461
	30	0	1 159	758 269	0	3 443	2 745 475
	32	746	3 359	1 866 914	-	-	-
	33	0	3 632	1 907 738	-	-	-

6 Conclusion

We have shown some new results on symmetry for finite model generation of many-sorted theories. The trivial symmetry is shown to be a particular case of this new symmetry. A symmetry detection algorithm is given. The trivial symmetry and the LNH heuristic are combined with the detected symmetry to maximize the gain. The method SEM augmented with the property of the detected symmetry is experimented on several problems. The experimental results obtained with our first implementation are satisfactory. Many improvements of the implementation are possible. One can refine the necessary symmetry conditions to increase the efficiency of symmetry detection. Another point of interest is to find some other trivial symmetries which do not need detection. Our experiments are still in progress. We also aim to apply our results for finite model generation of modal and temporal logics.

References

1. Gilles Audemard and Laurent Henocque. The extended least number heuristic. In *Proceedings of IJCAR*, Springer Verlag, June 2001.
2. Belaid Benhamou and Laurent Henocque. A hybrid method for finite model search in equational theories. *Fundamenta Informaticae*, 39(1-2):21–38, 1999.
3. Belaid Benhamou and Lakhdar Sais. Tractability through symmetries in propositional calculus. *Journal of Automated Reasoning*, 12(1):89–102, 1994.
4. James Crawford. A theoretical analisys of reasoning by symmetry in first order logic. In *Proceedings of Workshop on Tractable Reasoning*, pages 17–22, 1992.

Table 3. Rings - Comparison.

		SEM with detection			SEM without detection		
Pb	Size	Nb Models	Time	Nodes	Nb Models	Time	Nodes
RU	14	1	11	9 184	1	21	25 421
	16	335	50	49 914	1 745	162	51 327
	18	4	180	100 523	19	848	218 955
	20	4	683	217 502	21	5977	2 343 901
	21	1	854	237 697	-	-	-
	23	1	1 408	261 957	-	-	-
	25	9	3 665	603 360	-	-	-
	27	80	4 721	656 648	-	-	-
RNA	10	0	62	252 720	0	55	252 720
	12	0	231	504 258	0	209	504 258
	14	0	432	595 340	0	592	595 340
	15	0	621	646 421	0	592	646 421
RNB	36	0	613	209 291	0	3 832	1 916 142
	38	0	768	209 291	0	4 872	1 916 142
	40	0	957	209 291	0	6 178	1 916 142
	41	0	1 066	209 291	0	6 871	1 916 142
	42	0	1 163	209 291	-	-	-
	43	0	1 279	209 291	-	-	-
	44	0	1 420	209 291	-	-	-
	46	0	1 705	209 291	-	-	-

5. James Crawford, Matthew L. Ginsberg, Eugene Luck, and Amitabha Roy. Symmetry-breaking predicates for search problems. In *proceedings of KR'96*, pages 148–159, 1996.
6. Masayuki Fujita, John Slaney, and Franck Bennett. Automatic generation of some results in finite algebra. In *Proceedings of IJCAI*, pages 52–57, 1993.
7. Nicolas Peltier. A new method for automated finite model building exploiting failures and symmetries. *Journal of Logic and Computation*, 8(4):511–543, 1998.
8. Christian B. Suttner and Geoff Sutcliffe. The TPTP Problem Library - v2.1.0. Technical Report JCU-CS-97/8, James Cook University, 1997.
9. Jian Zhang. Problems on the Generation of Finite Models. In *proceedings of CADE 12*, pages 753–757. Springer-Verlag 1994.
10. Jian Zhang. Constructing finite algebras with FALCON. *Journal of Automated Reasoning*, 17(1):1–22, 1996.
11. Jian Zhang and Hantao Zhang. SEM: a system for enumerating models. In *Proceedings of IJCAI*, pages 298–303, 1995.