

A New Set of Algebraic Benchmark Problems for SAT Solvers

Andreas Meier¹ and Volker Sorge²

¹ DFKI GmbH, Saarbrücken, Germany
ameier@dfki.de

² School of Computer Science, University of Birmingham, UK
V.Sorge@cs.bham.ac.uk

Abstract. We propose a new benchmark set consisting of problems generated during the construction of classification theorems for quasigroups. It extends and generalises the domain of quasigroup existence problems, to which SAT solvers have been applied successfully in the past, to a rich class of benchmarks of varying difficulty.

1 Introduction

Classification of mathematical structures is a challenging task in mathematical research. A first step in producing algebraic classification theorems is to determine for which sizes certain algebras exist. Automated support for solving such existence problems has been remarkably successful in the past. In particular, model generators [4, 8] and satisfiability (SAT) solvers [15, 14] have been successfully applied to show the existence or non-existence of quasigroups with particular associated properties. The study and discovery of quasigroups with certain properties is interesting also outside of pure mathematics, because the underlying structure of quasigroups is similar to that found in many real-world applications [6, 7]. While quasigroup existence problems present a challenging task for SAT solvers their use as a benchmark set is restricted as, firstly, the number of problems that can be tackled realistically by existing systems is relatively small and, secondly, the structure of the problems is relatively similar.

In [3] we have extended quasigroup existence problems to the more general classification problem: How can different isomorphism classes of quasigroups of a given cardinality be described by their algebraic properties? The reasoning problems that need to be solved when answering this question are generally of propositional nature and can effectively be transformed into SAT problems [9].

In this paper, we present a uniform view of the classification problems as generalised quasigroup existence problems and suggest them as a new set of benchmarks from an algebraic domain for the testing and development of SAT solvers. The classification process can be seen as generator for this new benchmark set containing problems of varying structure and complexity (Sec. 2). We describe some of the characteristics of the problems with respect to their encoding in propositional logic (Sec. 3) and present a non-trivial example to give an impression of the mathematics behind the domain (Sec. 4).

2 Quasigroup Classification Problems

In [3] we have presented a bootstrapping algorithm to automatically generate classification theorems in finite algebra. To guarantee correctness of the classification a number of theorems has to be shown. Each theorem is essentially an existence problem and can be tackled by SAT solvers. We can therefore view the overall bootstrapping algorithm as a generator of our benchmark set.

The *general existence problem in finite algebra* is the question whether for some cardinality n an algebra exists that satisfies a given set of axioms \mathcal{A} . A considerable amount of research in automated reasoning has been devoted to solving some existence problems for the particular domain of quasigroups. A quasigroup is a non-empty set Q together with a binary operation \circ that satisfies the property $\forall a, b \in Q. (\exists x \in Q. x \circ a = b) \wedge (\exists y \in Q. a \circ y = b)$. This property is often called Latin Square property and has the effect that every element of Q appears exactly once in every row and every column of the multiplication table of \circ . One existence problem for quasigroups is, for instance, to ask whether a quasigroup of a given cardinality n exists for which the operation \circ also satisfies the QG4-property $\forall x, y \in Q. (x \circ (x \circ y)) \circ y = x$. SAT solvers could show, for example, that a QG4-quasigroup of cardinality $n = 14$ exists [15].

A *classification problem in finite algebra* for a given cardinality n and a set of axioms \mathcal{A} is the question: How many different algebras exist that satisfy \mathcal{A} and how can they be described? By “different” we mean here that the algebras are not isomorphic to each other; thus the classification problem is actually concerned with the detection and the axiomatic description of all isomorphism classes of algebras satisfying the axioms \mathcal{A} . In [3] we tackle this problem employing a bootstrapping algorithm. It automatically constructs a decision tree by successively identifying non-isomorphic structures and their discriminating properties until all possible isomorphism classes are generated and for each isomorphism class a *representant* (i.e., an algebra that is element of the isomorphism class) is found. A property P acts as a *discriminant* for any two algebras A and B iff $P(A)$ and $\neg P(B)$ means that A and B are not isomorphic. An isomorphism class can be uniquely described by its associated discriminants that form a *classifying property*, that is a property that holds for every algebra in the isomorphism class, but does not hold for any algebra outside the isomorphism class.

For example, Fig. 1 contains the decision tree for the classification problem of order 3 quasigroups together with the representants for each of the 5 isomorphism classes, where representant A_i belongs to the node i in the tree. The edges of the decision tree are labelled with discriminants and each node in the tree is associated with a *describing property*, which is the conjunction of all discriminants along the path from the root to the node. The nodes associated with isomorphism classes are the leaf nodes marked with a double circle. The classifying property of an isomorphism class is the describing property of the associated node. For instance, the isomorphism class represented by node 6 has the classifying property of $P_1 \wedge \neg P_2 \wedge P_3 \wedge P_4$. If during the construction of the decision tree a node is reached for which no representant exists that satisfies the

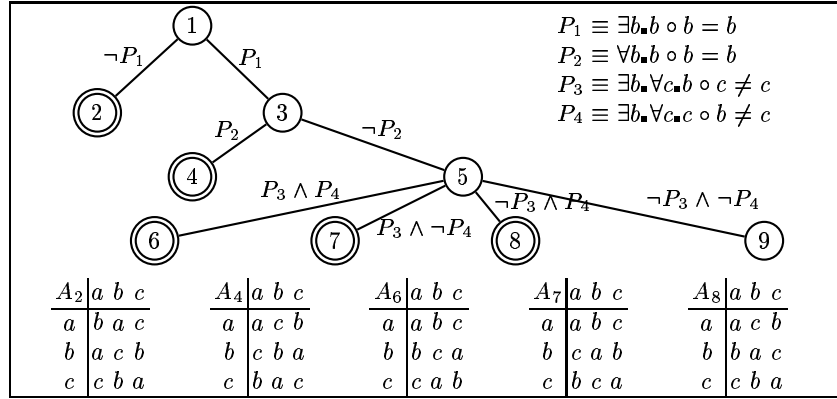


Fig. 1. Decision tree and isomorphism class representants for order 3 quasigroups.

describing property of the node, then this node is a dead-end in the tree. Node 9 in the tree in Fig. 1 is an example of such a node.

To verify the correctness of the classification the properties of the decision tree have to be proved. Thereby, the most difficult problem is to show that actually a leaf node is reached, i.e., that a node either represents an isomorphism class or that it is a dead-end. In the latter case we show that no algebra of cardinality n satisfies the describing property of the node (we call this the Dead-End Theorem), which is obviously an existence problem. In the former case we have to show that all algebras of cardinality n that satisfy the describing property of the node are isomorphic (Isoclass Theorem). This is also an existence problem, since it states that there does not exist an algebra A that satisfies the describing property and is not in the isomorphism class.

Although the isoclass theorems are essentially second order, they can be expressed as propositional logic problems by enumerating all possible isomorphism functions for structures of cardinality n . The naïve approach, considering all $n!$ possible isomorphisms, can be improved upon by first proving a lemma that states that all algebras with cardinality n satisfying a property P share a generating system¹. Since generating systems are invariant under isomorphism (i.e., if algebra A has a generating system and algebra B is isomorphic to A , then B has also such a generating system) it is a necessary prerequisite that this lemma holds for the isomorphism-class theorem to hold. We call this lemma the *generating-system lemma*. If it can be shown, then the number of isomorphisms for the isoclass theorem can often be drastically reduced to only the possible mappings of the generators. The generating-system lemma is again an existence problem stating that there does not exist an algebra A that satisfies classifying property P , but does not exhibit a certain generating system.

¹ A structure A with binary operation \circ is said to be *generated* by a set of elements $\{a_1, \dots, a_m\} \subseteq A$ if every element of A can be expressed as a combination – usually called a factorisation or word – of the a_i under the operation \circ . We call a set of generators together with the corresponding factorisations a *generating system*.

3 The Benchmark Set and its Characteristics

The *problem set* consists of a collection of generating-system lemmas, isoclass theorems, and dead-end theorems for quasigroups of cardinality 6, 7, and 8. The problems have been generated from decision trees for the classification of

- non-idempotent quasigroups of cardinality 6 with a simplified version of the QG3 property: $\exists a, \forall b, (b \circ a) \circ (a \circ b) = b$,
- quasigroups 7 with the QG9 property: $\forall a, b, x, y, a \circ b = x \wedge x \circ b = y \rightarrow y \circ b = a$,
- quasigroups 8 with a simplified version of QG9: $\forall a, b, x, a \circ b = x \rightarrow x \circ b = a$.

The current set contains roughly 1000 problems and has been submitted in the DIMACS format as a benchmark set to the SAT05 system competition. The set comprises both satisfiable and unsatisfiable problems. This has essentially two reasons: Firstly, during the construction of the decision tree, generating-system and isoclass theorems are only shown for leaf nodes associated with isomorphism classes. If the tree has been constructed correctly these theorems hold and the corresponding SAT problems are unsatisfiable. However, for the benchmark set we also conjecture generating-system lemmas and isoclass theorems for branching nodes, with respect to the describing properties of that node. For a branching node, however, either one or both of the conjectures can not hold, which leads to satisfiable problems. Secondly, the decision trees for the quasigroups 7 and 8 problems are so far only partially constructed and therefore the status of some of the resulting SAT problems is not yet known.

Our *encoding* follows Zhang’s approach [13], where a boolean variable corresponds to an equation of the form $x \circ y = z$ or $x = y$, where x, y, z are instantiated with the n elements of the algebra of cardinality n . Hence, to encode all possible equations $n^3 + n^2$ boolean variables are necessary, such that the number of boolean variables increases wrt. the cardinality n (see [9] for details).

Since the discriminant properties can be arbitrarily complex and nested it turns out that a naïve clause normalisation approach suffers in our domain from a combinatorial explosion of the number and the length of the resulting clauses. Hence, we adopted clause normalisation techniques from [11] that aim to create small clause normal forms. The basic technique is the introduction of additional boolean variables to suitably break formulas. This avoids combinatorial explosion and restricts the size of the resulting clauses but extends the problem formalisation by additional boolean variables. Our clause normalisation breaks formulas when the resulting clauses become larger than $3 \cdot n$. The introduction of additional boolean variables to break the formulas, however, can result in final clauses that are slightly larger than $3 \cdot n$. The optimal clause normalisation for our problems is still subject to testing.

The *difficulty* of propositional satisfiability problems can generally be characterised both by the number of boolean variables as well as by the number and sizes of the formulas or clauses involved. In our domain, both the number of variables and clauses increase for larger cardinalities n and for nodes deeper in the decision tree. To illustrate this consider the figures characterising

Cardinality	depth of node	#variables	#clauses	#max clause size
5	1	150	2055	5
5	23	352	4531	9
6	1	257	5925	6
6	7	405	6975	8
7	1	2401	20594	7
7	8	4044	24907	12
8	1	576	53212	7
8	8	12671	110508	28

Table 1. Characteristic numbers of quasigroup generating-system lemma problems.

generating-system lemmas of quasigroup classification problems with cardinality 5–8 in Table 1.

We have applied the SAT solvers zChaff [10], DPLL T [5], CVC lite [2] and the first-order theorem prover Spass [12] to different formalisations of our benchmark problems (see [9] for details). These experiments reveal that on average the difficulty of the problems does indeed increase with increasing numbers of variables and clauses. However, they also show that there is a high variance in the time needed by the systems to solve problems of roughly the same size. While some problems could be solved very quickly, other problems with a similar number of variables and clauses would take very long to solve or could not be solved at all. In fact, the complexity of the clauses for a problem in our domain depends mainly on the form of the describing property P , which is a conjunction of discriminant properties (see Sec. 2). In general, P is more complex the deeper the corresponding node is in the tree. However, the complexity of the discriminants of which P is composed can differ (see the example in the next section) and the difficulty they pose to SAT solvers can vary. For instance, the discriminant $\forall b. b \circ b = b$ simplifies the problem at hand as opposed to a property like the quasigroup property itself, which makes the problem generally more difficult.

For similar reasons isoclass theorems are generally considerably less difficult than dead-end theorems and generating-system lemmas. The usage of the generating-system lemma as axiom in the isoclass theorem drastically reduces the complexity of this theorem, since the generating-system lemma results in unit clauses restricting the search. And our experiments indeed established the fact that the generating-system lemmas are the hardest problems of our domain.

4 An Example Problem

To give an impression of the mathematical quality of the problems in our benchmark set we present a non-trivial example from our domain. For the classification of quasigroups of order 7 with the QG9 property our algorithm computes an isomorphism class described by the representant A below together with a classifying property consisting of the conjunction of the following 11 properties:

A	e_1	e_2	e_3	e_4	e_5	e_6	e_7	
e_1	e_4	e_1	e_5	e_7	e_6	e_2	e_3	1. $\forall b, b \circ b \neq b$
e_2	e_1	e_6	e_3	e_4	e_5	e_7	e_2	2. $\exists b, \forall c, b \circ c \neq c$
e_3	e_7	e_4	e_6	e_2	e_1	e_3	e_5	3. $\exists b, c, b \circ c = b \wedge c \circ b = b$
e_4	e_2	e_7	e_1	e_3	e_4	e_5	e_6	4. $\forall b, \exists c, d, c \circ d = b \wedge d \circ c \neq b$
e_5	e_3	e_2	e_4	e_5	e_7	e_6	e_1	5. $\exists b, \forall c, (c \circ b) \circ c \neq b$
e_6	e_6	e_5	e_2	e_1	e_3	e_4	e_7	6. $\exists b, \forall c, b \circ (b \circ c) \neq c$
e_7	e_5	e_3	e_7	e_6	e_2	e_1	e_4	7. $\forall b, \exists c, b \circ (b \circ c) \neq c$
								8. $(\exists b, \forall c, (b \circ c) \circ (b \circ c) \neq c) \wedge$ $(\forall b, c, b \circ b \neq c \vee c \circ c \neq b)$
								9. $\exists b, c, b \circ b = c \wedge b \circ c = b$
								10. $\exists b, \forall c, (b \circ c) \circ b \neq c$
								11. $\exists b, \forall c, c \circ (c \circ b) \neq b \vee b \circ c = c \circ b$

For the representant A the algorithm computes a generating system, whose set of generators consists of e_7 only and the factorisations are:

$$\begin{aligned}
e_1 &= ((e_7 \circ e_7) \circ (e_7 \circ e_7)) \circ (((e_7 \circ e_7) \circ (e_7 \circ e_7)) \circ e_7) & e_4 &= e_7 \circ e_7 \\
e_2 &= (e_7 \circ (e_7 \circ e_7)) \circ ((e_7 \circ e_7) \circ (e_7 \circ e_7)) & e_5 &= ((e_7 \circ e_7) \circ (e_7 \circ e_7)) \circ e_7 \\
e_3 &= (e_7 \circ e_7) \circ (e_7 \circ e_7) & e_6 &= e_7 \circ (e_7 \circ e_7)
\end{aligned}$$

The generating-system lemma for this isomorphism class states that all algebras of order 7 that satisfy the given classifying property exhibit a generating system of the above form, i.e., they exhibit exactly this generating system or a generating system resulting from the permutation of the elements $e_1, e_2, e_3, e_4, e_5, e_6, e_7$. This results in $7!$ concrete possible generating systems, which have to be encoded in propositional logic. The final SAT problem then consists of 972 boolean variables and 13573 clauses with a maximal clause length of 13 literals.

If the generating-system lemma is shown successfully, the corresponding isoclass theorem can be stated as: All algebras of order 7 that satisfy the given classifying property and have a generating system of the above form are isomorphic to A . The possible isomorphisms that need to be considered are now restricted to the set of generators. Since we only have one generator, there are 7 possible isomorphisms, which can be further restricted by discarding those mappings that would violate the homomorphism property (see [9] for details). For the problem at hand it turns out that there is exactly one possible isomorphism on the generating system that needs to be instantiated into the isoclass theorem. The concrete SAT problem then consists of 1122 boolean variables and 8662 clauses with a maximum clause length of 14 literals.

Although the pure numbers of variables and clauses might suggest that the isoclass theorem and the generating-system lemma are approximately of the same complexity, the actual complexity is hidden in the generating-system lemma (see discussion in previous section). And indeed when applying zChaff to the problems it proves the isoclass theorem in 0.1 seconds but needs 19154.9 seconds to show the generating-system lemma. The other systems we employed in our experiments exhibited the same drastic difference in performance.

5 Conclusions

We have presented the domain of classification problems for quasigroups and suggest it as a new set of benchmarks for satisfiability solving. We believe that

using these problems for the development and testing of SAT solvers can be of mutual benefit: On the one hand, the SAT community will gain a large testbed of problems of varying difficulty and structure. On the other hand stronger and better SAT solvers can aid in the derivation of new mathematical classification results. While we have concentrated on quasigroups for the submitted benchmark set, our classification algorithm has already been applied to other algebraic structures, such as monoids and semi-groups, which might also be exploitable in the future.

The experience we have gained so far suggests that the successful application of SAT solvers in our domain relies on an effective clause normalisation and a suitable formalisation of properties. Indeed, in our experiments different formalisations of properties (e.g., the quasigroup property) had also a considerable impact on the difficulty of the resulting SAT problems. However, it is not yet clear how, in general, properties of our domain can be best reformulated or encoded to enhance the performance of a SAT solver. Further experiments with large numbers of properties should give us more insights into these questions. These might also prove beneficial for other, possibly non-mathematical domains.

References

1. R. Alur, D. Peled, editors. *Proc. of CAV-2004*, LNCS 3114. Springer, 2004.
2. C. Barrett, S. Berezin. CVC Lite: A new implementation of the cooperating validity checker. In Alur and Peled [1], p.515–518.
3. S. Colton, A. Meier, V. Sorge, R. McCasland. Automatic generation of classification theorems for finite algebras. In *Proc. of IJCAR-2*, LNAI 3097, p.400–414. Springer, 2004.
4. M. Fujita, J. Slaney, F. Bennett. Automatic Generation of Some Results in Finite Algebra. *Proc. of IJCAI-13*, p.52–57. Morgan Kaufmann, 1993.
5. H. Ganzinger, G. Hagen, R. Nieuwenhuis, A. Oliveras, C. Tinelli. Dpll(t): Fast decision procedures. In Alur and Peled [1], p.175–188.
6. S.R. Kumar, A. Russel, R. Sundaram. Approximating latin square extensions. *Algorithmica*, 24:128–138, 1999.
7. C. Laywine, G. Mullen. *Discrete Mathematics using Latin Squares*. Wiley, 1998.
8. W.W. McCune. A davis-putnam program and its application to finite first-order model search: quasigroup existence problems. Report, Argonne Nat. Labs, 1994.
9. A. Meier, V. Sorge. Applying sat solving in classification in finite algebra. Submitted to the Journal of Automated Reasoning.
10. M. Moskewicz, C. Madigan, Y. Zhao, L. Zhang, S. Malik. Chaff: Engineering an efficient sat solver. In *Proc. of the Design Automation Conference*, p.530–535, 2001.
11. A. Nonnengart, C. Weidenbach. Computing small clause normal forms. In *Handbook of Automated Reasoning*. Elsevier, 2001.
12. C Weidenbach, U Brahm, T Hillenbrand, E Keen, C Theobald, D Topic. SPASS version 2.0. In *Proc. of CADE-18*, LNAI 2392, pages 275–279. Springer, 2002.
13. H. Zhang. Specifying latin squares in propositional logic. In *Automated Reasoning and Its Applications, Essays in honor of Larry Wos*. MIT Press, 1997.
14. H. Zhang, M. Bonacina, J. Hsiang. PSATO: a distributed propositional prover and its application to quasigroup problems. *J. of Symb. Computation*, 21:543–560, 1996.
15. H. Zhang, J. Hsiang. Solving Open Quasigroup Problems by Propositional Reasoning. In *Proc. of International Computer Symposium*, 1994.