Computer Search for Large Sets of Idempotent Quasigroups*

Feifei Ma[1,2] and Jian Zhang[1]

[1] State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences
[2] Graduate University, Chinese Academy of Sciences
{maff,zj}@ios.ac.cn

Abstract. A collection of $n-2$ idempotent quasigroups of order n is called a large set if any two of them are disjoint, denoted by $LIQ(n)$. While the existence of ordinary $LIQ(n)$ has been extensively studied, the spectrums of large sets of idempotent quasigroups with various identities remain open, for example, large set of Steiner pentagon quasigroups of order 11 which is denoted by $LSPQ(11)$. This paper describes some computer searching efforts seeking to solve such problems. A series of results are obtained, including the non-existence of $LSPQ(11)$.

1 Introduction

The quasigroup problem has long been the focus of much interest in combinatorics. Many classes of finite quasigroups attract such attention partly because they are very natural objects in their own right and partly because they are correlative to design theory. A number of hard combinatorial problems are also raised by quasigroups. Over the last decade, the study of quasigroups has largely benefited from the improvement of automated reasoning techniques. Many open problems, to which the conventional mathematical methods are hard to apply, have been solved by means of computer search. For example, a series of open problems of the type from QG2 to QG9 were settled by several model generators such as *MGTP,FINDER,SEM* and the propositional satisfiability prover *SATO,DDPP* respectively [4,8,12,10,11]. Later, the non-existence of QG2(10), which used to be quite difficult, was established by Dubois et al. with their specific-purpose program *qgs* [3]. In fact, quasigroups with certain identities are highly structured. It is natural to translate the constraints for a quasigroup to logic formulae or model it as a constraint satisfaction problem, which can be effectively handled by these automatic tools.

A more challenging problem arising in this field is the large set problem for various idempotent quasigroups. Unlike the above problems, we need to find a set of quasigroups satisfying certain constraints rather than only one. Apparently this is more difficult since the search space to be explored is exponentially

* This work is supported by the National Natural Science Foundation of China (NSFC) under grant No. 60673044.

D. Kapur (Ed.): ASCM 2007, LNAI 5081, pp. 349–358, 2008.
© Springer-Verlag Berlin Heidelberg 2008

more huge. Lie Zhu summarized some open cases of such problems in [13] and [14]. Using the first order model generator SEM together with a novel searching strategy, we are able to solve some of the problems. This paper gives a summary of the experimental results and a brief description of the search method.

2 Preliminaries

2.1 Definitions

Firstly, let us recall some notations:

Definition 1 (Quasigroup). *A* quasigroup *is an ordered pair* $(Q, \oplus)$*, where* Q *is a set and* $\oplus$ *is a binary operation on* Q *such that the equations* $a \oplus x = b$ *and* $y \oplus a = b$ *are uniquely solvable for every pair of elements* a*,* b *in* Q*.*

For a finite set Q, the order of the quasigroup $(Q, \oplus)$ is denoted as $|Q|$.

A quasigroup $(Q, \cdot)$ is *idempotent* if the identity $x \cdot x = x$ (briefly $x^2 = x$) holds for all x in Q. We denote an idempotent quasigroup of order n as $IQ(n)$.

Two idempotent quasigroups $(Q, \oplus)$ and $(Q, \cdot)$ are said to be *disjoint* if for any $x, y \in Q$, $x \oplus y \neq x \cdot y$ whenever $x \neq y$.

Definition 2 (Large Set). *A collection of idempotent quasigroups* $(Q, \oplus_1)$*,* $(Q, \oplus_2)$*, ...,* $(Q, \oplus_{n-2})$*, where* $n = |Q|$*, is called a* large set *if any two of the idempotent quasigroups are disjoint.*

A large set of idempotent quasigroups of order n is denoted by $LIQ(n)$.

Definition 3 (Steiner Pentagon Quasigroup). *A quasigroup of order* n *is called a* Steiner pentagon quasigroup *if it satisfies the identities* $\{x^2 = x, (yx)x = y, x(yx) = y(xy)\}$*, denoted by* $SPQ(n)$*.*

Obviously a $SPQ(n)$ is a particular kind of $IQ(n)$. A large set of Steiner pentagon quasigroups of order n is denoted by $LSPQ(n)$.

2.2 The Problems

The existence of $LIQ(n)$ has already been established by Teirlinck and Lindner [9], and Chang [1]. In [1], Chang concluded that there exists an $LIQ(n)$ for any $n \geq 3$ with the exception $n = 6$. However, for IQs with certain identities, the spectrum of large sets has not been explored extensively so far. Generally there are two classes of IQs with which we are concerned in the paper. The first class includes 7 kinds of idempotent quasigroups whose existence has been studied systematically. These quasigroups satisfy following identities, respectively:

1. $xy \cdot yx = x$ Schröder quasigroup
2. $yx \cdot xy = x$ Stein's third law
3. $(xy \cdot y)y = x$ C_3-quasigroup
4. $x \cdot xy = yx$ Stein's first law; Stein quasigroup
5. $(yx \cdot y)y = x$

6. $yx \cdot y = x \cdot yx$ Stein's second law
7. $xy \cdot y = x \cdot xy$ Schröder's first law

Any short conjugate-orthogonal identity, if nontrivial, is conjugate-equivalent to one of them[2]. The large set of idempotent quasigroups satisfying property (i) of order n is denoted by $LIQ^{(i)}(n)$. Since for some orders, there is no IQ, Lie Zhu [14] summarized the open cases in the following list. They are of the moderate orders, therefore may be suitable for computer search.

Table 1. Open Cases for LIQ of Moderate Sizes

1. $LIQ^{(1)}(8)$	$LIQ^{(1)}(12)$	$LIQ^{(1)}(13)$	
2. $LIQ^{(2)}(5)$	$LIQ^{(2)}(9)$	$LIQ^{(2)}(12)$	
3. $LIQ^{(3)}(4)$	$LIQ^{(3)}(7)$	$LIQ^{(3)}(10)$	$LIQ^{(3)}(13)$
4. $LIQ^{(4)}(4)$	$LIQ^{(4)}(5)$	$LIQ^{(4)}(9)$	$LIQ^{(4)}(11)$
5. $LIQ^{(5)}(5)$	$LIQ^{(5)}(7)$	$LIQ^{(5)}(8)$	$LIQ^{(5)}(11)$
6. $LIQ^{(6)}(5)$	$LIQ^{(6)}(9)$	$LIQ^{(6)}(13)$	
7. $LIQ^{(7)}(8)$	$LIQ^{(7)}(9)$	$LIQ^{(7)}(13)$	

The second class of IQs is SPQ, which we already mentioned. It is known [6,13] that the spectrum for Steiner pentagon quasigroups is precisely the set of all positive integers $n \equiv 1$ or 5 (mod 10), except for $n = 15$. Zhu pointed out that the smallest unknown order for LSPQ is 11.

3 Search for LIQs of Small Orders

SEM is a general-purpose search program for finding finite models. It accepts a set of first order formulae as input and tries to find one or a specified number of models. The size of the model should be given by the user. Since the search is exhaustive, when the program terminates without finding any model, it means that there is no model of the given size.

With the help of SEM, we are able to perform the search for $LIQ^{(i)}(n)$. Without loss of generality, we assume the domain Q to be the set $\{0,1,\ldots,n-1\}$. A quasigroup is actually a function $f : Q \times Q \mapsto Q$ satisfying the constraints

$$\forall x \forall y \forall z (f(x, y) = f(x, z) \to y = z)$$

and

$$\forall x \forall y \forall z (f(x, y) = f(z, y) \to x = z)$$

Hence to find a large set of quasigroups with some identity is to determine $n-2$ such functions, namely $f_1, f_2, \ldots, f_{n-2}$, with extra constraint of the identity, and what's more, the disjoint constraints:

$$\forall x \forall y (f_i(x, y) = f_j(x, y) \to x = y)$$

where $i \neq j$. All of these constraints, together with the functions, form the input file for SEM.

We completed all the searches for $LIQ^{(i)}(n)$s with n no more than 8. The experiments were performed on an Intel 1.86GHZ 2CPU PC with Fedora 7 OS. The results are listed in Table 2. For each of the order 8 cases, the execution time of SEM ranges from 3 seconds to 3 minutes; for all other cases, the execution time is much less than a second. For each case where there is no LIQ, we further obtained the maximum number of disjoint quasigroups, denoted by $D^{(i)}(n)$, which is also useful in mathematics. When $D^{(i)}(n) = 1$, there are no disjoint $IQ^{(i)}(n)$s while $IQ^{(i)}(n)$s do exist. We have used Mace4 [7] to double check the results.

Table 2. Search Results for $LIQ^{(i)}(n)$s with $n \leq 8$

Identity i	Order n	Existence of LIQ	$D^{(i)}(n)$
1	8	YES	-
2	5	NO	2
3	4	YES	-
	7	NO	2
4	4	YES	-
	5	NO	1
5	5	NO	1
	7	NO	1
	8	NO	3
6	5	NO	2
7	8	YES	-

4 Search for $LSPQ(11)$

Intuitively, it seems infeasible to complete the computer search for $LSPQ(11)$ using the direct method. The highest order of $LSPQ(n)$ that has been completed by SEM is 10, but for order 11 it is much more difficult. Our experience is that, even searching for a partial model of 4 disjoint $SPQ(11)$s would take as long as 4 hours without any result. Nevertheless, via a tactical utilization of SEM, we are able to conclude the non-existence of $LSPQ(11)$. The effectiveness of our approach resides essentially in two strategies:

1. Constraint weakening.
2. Isomorphism elimination.

We know a **permutation** of a Domain Q is a one to one mapping (bijection) from Q onto itself. Let us denote a first order theory, i.e., a set of first order formulae by Σ. For two models M_1 and M_2 of Σ on Q, if there exists a permutation P that maps one of them to the other, then M_1 and M_2 are **isomorphic**.

Lemma 1. *Given a first order theory Σ and a finite domain Q. If there is no element in Q appearing in Σ as a constant, then for any model M of Σ on Q and any permutation P on Q, the new interpretation $P(M)$ obtained by performing P on M is also a model of Σ on Q.*

The lemma is straightforward since the assumption guarantees the interchangeablity of all elements in Q. Consequently, we have the following theorem which serves as the foundation in our approach:

Theorem 1. *Let S_n be the set of non-isomorphic $SPQ(n)$s. An $LSPQ(n)$, if there exists any, is isomorphic to an $LSPQ(n)$ containing at least one $SPQ(n)$ in S_n.*

Proof. First of all let us represent all the constraints for $LSPQ(n)$ by first order formulae and collect them in the set $\Sigma_{LSPQ(n)}$. $\Sigma_{LSPQ(n)}$ should be of the following form:

$$\begin{aligned}\Sigma_{LSPQ(n)} = \{ & \bigwedge_{1\le i\le n-2} \forall x \forall y \forall z (f_i(x,y) = f_i(x,z) \rightarrow y = z), \\ & \bigwedge_{1\le i\le n-2} \forall x \forall y \forall z (f_i(x,y) = f_i(z,y) \rightarrow x = z), \\ & \bigwedge_{1\le i\le n-2} \forall x f_i(x,x) = x, \\ & \bigwedge_{1\le i\le n-2} \forall x \forall y f_i(f_i(y,x),x) = y, \\ & \bigwedge_{1\le i\le n-2} \forall x \forall y f_i(x, f_i(y,x)) = f_i(y, f_i(x,y)), \\ & \bigwedge_{1\le i<j\le n-2} \forall x \forall y (f_i(x,y) = f_j(x,y) \rightarrow x = y)\}\end{aligned}$$

Apparently $\Sigma_{LSPQ(n)}$ is a first order theory with no constant appearing in its formulae. Suppose there exists an $LSPQ(n)$, namely L. So L is a model of $\Sigma_{LSPQ(n)}$. Arbitrarily choose an $SPQ(n)$ from L, it must be isomorphic to some $SPQ(n)$ A in S_n since all non-isomorphic $SPQ(n)$s are included in S_n. Denote the permutation which maps the chosen one in L to A by P. Perform the permutation P on L and denote the result by $P(L)$, by lemma 1 $P(L)$ is a model of $\Sigma_{LSPQ(n)}$, i.e., an $LSPQ(n)$. What's more, A is contained in $P(L)$ because it is obtained by performing P on the originally chosen $SPQ(n)$, which is part of L. By definition L is isomorphic to $P(L)$, hence the theorem holds. □

Furthermore, it's observed that a large set of idempotent quasigroups has the following property:

Lemma 2. *Suppose an $LIQ(n)$=$\{(Q, \oplus_1), (Q, \oplus_2), \ldots, (Q, \oplus_{n-2})\}$, where $n = |Q|$. For any $x, y \in Q$, if $x \neq y$, then the collection of $x \oplus_i y$ is exactly the set Q excluding x and y, or formally, $\{x \oplus_i y | 1 \le i \le n-2\}$=$Q$-$\{x,y\}$.*

Proof. For any $1 \leq i \leq n-2$, from the idempotent identity we have $x \oplus_i x = x$ and $y \oplus_i y = y$. Since $(Q, \oplus_i)$ is a quasigroup and $x \neq y$, we have $x \oplus_i y \neq x$, otherwise the equation $x \oplus_i z = x$ with the unknown z is not uniquely solvable. Similarly, we get $x \oplus_i y \neq y$. Therefore, $x \oplus_i y \in Q - \{x, y\}$. Also, the disjoint property of large set implies that for any $i \neq j$, we have $x \oplus_i y \neq x \oplus_j y$. So the cardinality of $\{x \oplus_i y | 1 \leq i \leq n-2\}$ is $n-2$, equaling to the cardinality of Q-{x,y}. The two sets are equal. □

Now we explain the basic idea of our method. Suppose there are m non-isomorphic $SPQ(n)$s in total, namely $A_1, A_2, \ldots, A_m$. By Theorem 1 we know that $LSPQ(n)$ exists if and only if there is an $LSPQ(n)$ containing at least one such A_i. Therefore without losing any non-isomorphic solution, we can safely divide the search space into m (maybe intersecting) sub-spaces, each of which corresponds to an $A_i \in LSPQ(n)$. Now let's consider any of the m situations. While searching in the ith sub-space, the first $SPQ(n)$ is prefixed to be A_i. We are to determine the multiplication tables of the rest $n-3$ $SPQ(n)$s in the large set. We denote them by $f_2^i, f_3^i, \ldots, f_{n-2}^i$. For these $n-3$ SPQs, it doesn't make any difference how they are ordered. We can choose a cell which is not on the diagonal, for example, $cell(0,1)$, and sort them by the cell value. Lemma 2 implies that any $f_j^i(0,1)$ and $f_k^i(0,1)$ are different for $j \neq k$, thus we can fix

$$f_2^i(0,1) < f_3^i(0,1) < \ldots < f_{n-2}^i(0,1) \tag{1}$$

and $(n-3)!-1$ isomorphisms are then eliminated. Also, if $n-3$ $SPQ(n)$s in the large set are determined, the candidate for the last one is unique and can be worked out immediately. This is because by Lemma 2, there is only one candidate value left for each $cell(x,y)$ where $x \neq y$, and the idempotent property forces each $cell(x,x)$ to be assigned x. What we need to do is to check if the candidate satisfies the identities of SPQ and it is quite an easy job. So the task is reduced to finding $n-4$ disjoint $SPQ(n)$s which are all disjoint with A_i, and satisfying (1).

Although the problem is much simplified, it remains intractable for the open case of order 11. We once tried one such subcase. SEM did not complete the search after running a week, and finally the process was killed. However, it is noticed that search for only one $SPQ(11)$ that is disjoint with A_i could be completed quite fast by SEM and it seems feasible to find all the solutions. If we weaken the disjoint constraints for f_j^i to be disjoint with A_i only, we can obtain the candidate set for f_j^i. Once all the candidates for $f_2^i, \ldots, f_{n-3}^i$ are found out, we can check if a large set can be formed.

Our method can be summarized as the following procedure:

1. Find the set of all non-isomorphic $SPQ(n)$s: $S_n = \{A_1, A_2, \ldots, A_m\}$. Currently we use SEMD [5].
2. For each A_i in S_n, do the following:
   2a Fix f_1^i to be A_i. For each f_j^i $(2 \leq j \leq n-3)$ use SEM to find all the candidates satisfying the following constraints:
      i. The SPQ(n) identities.
      ii. f_j^i(0,1) equals to the (j–1)th smallest value in the set Q–$\{0, 1, f_1^i(0,1)\}$.

iii. Disjoint with f_1^i.

and denote the candidate set by C_j^i.

2b Call the function `SCAN(i,n)` to find all the sets of $n-4$ SPQs which are disjoint with each other. If there are no such SPQs, try next A_i.

2c For each set of the disjoint SPQs calculate the possible solution for f_{n-2}^i and check if it is an $SPQ(n)$. If so, an $LSPQ(n)$ is discovered. If all the sets fail to produce an $LSPQ(n)$, try next A_i.

The program `SCAN(i,n)` is illustrated in Figure 1. It is a depth-first search program to find all combinations of $n-4$ disjoint SPQs. The $n-4$ SPQs are from $C_2^i, C_3^i, \ldots, C_{n-3}^i$ separately. The program works by scanning the SPQs in C_j^i $(2 \leq j \leq n-3)$ in sequence so as to pair up disjoint SPQs. The variable `cur` is the current search depth, i.e, the number of the candidate set being scanned. The current search path is kept in the array `pnt`. `C_i_j` is the array to store the SPQs in the candidate set C_j^i and `size[j]` is the cardinality of C_j^i. Initially `cur` is set to 3 and `pnt[j]` is set to 1.

If all $SPQ(n)$s in S_n have been tried and no $LSPQ(n)$ is found, the process terminates. Since the process is exhaustive, the non-existence of $LSPQ(n)$ can be established.

```
bool SCAN(i,n){
  sol_num=0;
  cur=3;
  while(TRUE){
    if(there exists some j, 1<j<cur, such that C_i_j[pnt[j]]
       and C_i_cur[pnt[cur]] are not disjoint)
       pnt[cur]++;
    else {if(cur==n-3){
             record pnt;
             sol_num++;
             pnt[cur]++;
           }
           else {cur++;
                  pnt[cur]=1;
                }
     }
    while(pnt[cur]==size[cur]+1){
       cur--;
       if(cur<2){
          if(sol_num==0) return FALSE;
          return TRUE;
       }
       pnt[cur]++;
     }
  }
}
```

Fig. 1. The program `SCAN(i,n)`

```
A1 |  0  1  2  3  4  5  6  7  8  9 10
---+---------------------------------
 0 |  0  2  6  5 10  9  8  1  7  4  3
 1 |  3  1  5  8  9  6  4  0 10  2  7
 2 |  4  0  2  6  7  3 10  9  5  1  8
 3 |  1  6  7  3  5  2  9  8  4 10  0
 4 |  2  7  8  9  4 10  1  5  3  0  6
 5 |  6 10  1  0  3  5  7  4  2  8  9
 6 |  5  3  0  2  8  1  6 10  9  7  4
 7 |  9  4  3 10  2  8  5  7  0  6  1
 8 | 10  9  4  1  6  7  0  3  8  5  2
 9 |  7  8 10  4  1  0  3  2  6  9  5
10 |  8  5  9  7  0  4  2  6  1  3 10
```

```
A2 |  0  1  2  3  4  5  6  7  8  9 10
---+---------------------------------
 0 |  0  2  7  5  6  8  9 10  1  3  4
 1 |  3  1  5  6  8  9  2  4  0 10  7
 2 |  4  0  2  8  5  3  1  6 10  7  9
 3 |  1  4  9  3 10  2  8  5  7  0  6
 4 |  2  3  8  9  4 10  7  1  6  5  0
 5 |  6  7  1  0  2  5 10  3  9  4  8
 6 |  5  9 10  1  0  7  6  2  4  8  3
 7 |  8  5  0 10  9  6  4  7  3  2  1
 8 |  7 10  4  2  1  0  3  9  8  6  5
 9 | 10  6  3  4  7  1  0  8  5  9  2
10 |  9  8  6  7  3  4  5  0  2  1 10
```

Fig. 2. Two non-isomorphic $SPQ(11)$s

Following the instructions above, we performed the search for $LSPQ(11)$. Firstly, there are only two non-isomorphic $SPQ(11)$s found by SEM, denoted by A_1 and A_2 respectively. They are shown in Fig. 2.

The search for C_j^i, $1 \leq i \leq 2$, $2 \leq j \leq 8$ is performed on an IBM BladeCenter with 8 2.5GHz PowerPC 970 2CPU processors. The search results are listed in Table 3. The times are also given in seconds. From Table 3 we can see that the

Table 3. Experimental Result for C_j^i

C_j^i	j:	2	3	4	5	6	7	8
$i=1$	Solution Number	1055	940	980	979	1055	929	924
	Running time	3593.13	3046.69	3362.07	3448.14	3590.43	3207.82	2994.13
$i=2$	Solution Number	830	732	758	732	726	867	804
	Running time	3614.09	2893.40	3418.53	3254.60	3374.20	3362.34	3167.44

cardinality of each set C_j^i is about 1000 and the running time doesn't exceed one hour in general.

For each of the two subspaces, we use the program in Fig. 1 to find 7 disjoint $SPQ(11)$s from the candidate sets. After running about 1 minute on the PC, the program returned `FALSE` for both cases, implying that $LSPQ(11)$ doesn't exist.

5 Discussion

It is interesting to notice that adding some redundant lemmas can greatly influence the running time of SEM. A remarkable evidence is the self-orthogonal property of SPQ. An idempotent quasigroup $(Q, \oplus)$ is called **self-orthogonal** if

$$\{(x \oplus y, y \oplus x) | x, y \in Q, x \neq y\} = \{(u, v) | u, v \in Q, u \neq v\}.$$

The property can be represented by the conjunction of the following two first order formulae.

$$\forall x \forall y (x \neq y \rightarrow f(x, y) \neq f(y, x)) \tag{2}$$

$$\forall x \forall y \forall z \forall w (x \neq y \wedge z \neq w \wedge (x \neq z \vee y \neq w) \rightarrow f(x,y) \neq f(z,w) \vee f(y,z) \neq f(w,z)) \quad (3)$$

It is known that an $SPQ(n)$ is self-orthogonal. Formula 2 can reduce the running time greatly in our experiments. We once tried to search for $LSPQ(9)$ directly by SEM. It lasted for about 10 hours and still could not give a result. By contrast, when formula 2 is added to the input file, the search is completed within a second, confirming the non-existence of $LSPQ(9)$. It is also helpful to the generating process of C_j^i, reducing the running time from almost 24 hours to less than an hour. However, when formula 3 is also added, in both cases the search would slow down dramatically, almost as slow as when there are no lemmas. While short lemmas are helpful, long ones may play a negative role. This issue may deserve further investigation.

Although the method in section 4 is designed for LSPQ, it can be generalized to $LIQ^i(n)$s or any other LIQs with similar identities. In addition, it can be easily modified to find out the maximum disjoint IQs if necessary. The basic idea is also suggestive to the automated search for other algebraic structures. We have just applied the method to $LIQ^2(9)$, $LIQ^4(9)$, $LIQ^6(9)$ and $LIQ^7(9)$ and established their nonexistences. It is hopeful that more open problems regarding the existence of large set can be solved in the future.

6 Conclusion

Searching for large set of idempotent quasigroups (LIQs) presents new challenges to computer scientists and mathematicians. The search space is much larger in general. This paper presents some search results and techniques which appear to be effective on the problem. The preliminary results are quite interesting to Lie Zhu and provided useful information for his research.

To reduce the search time, we have used three techniques:(1) adding redundant formulae to the original set of constraints; (2) dividing the set of constraints into several subsets, and constructing a large set gradually by considering various pairs of quasigroups; (3) employing the symmetry among the quasigroups in the large set, in addition to the symmetry among domain elements.

While the above techniques have helped us to obtain some results on previously open cases, we are also developing other techniques and investigating more difficult cases. Moreover, we will use other model finding tools in the future.

Acknowledgement

The authors are very thankful to Lie Zhu for his introduction to the problems.

References

1. Chang, Y.: The Spectrum for Large Sets of Idempotent Quasigroups. Journal of Combinatorial Designs (2000)
2. Colbourn, C.J., Dinitz, J.H.: Handbook of Combinatorial Designs, 2nd edn. CRC Press, Boca Raton (2007)

3. Dubois, O., Dequen, G.: The Non-existence of (3,1,2)-Conjugate Orthogonal Idempotent Latin Square of Order 10. In: Proc. of the 7th International Conference on Principles and Practice of Constraint Programming (2001)
4. Fujita, M., Slaney, J.K., Bennett, F.: Automatic Generation of Some Results in Finite Algebra. In: Proc. IJCAI, pp. 52–59 (1993)
5. Jia, X., Zhang, J.: A Powerful Technique to Eliminate Isomorphism in Finite Model Search. In: Proc. IJCAR, pp. 318–331 (2006)
6. Lindner, C.C., Stinson, D.R.: Steiner Pentagon Systems. Discrete Math. 52, 67–74 (1984)
7. McCune, W.: Mace4 Reference Manual and Guide. Technical Memorandum 264, Argonne National Laboratory, Argonne, IL, USA (2003)
8. Slaney, J., Fujita, M., Stickel, M.: Automated Reasoning and Exhaustive Search: Quasigroup Existence Problems. Computers and Mathematics with Applications 29, 115–132 (1995)
9. Teirlinck, L., Lindner, C.C.: The Construction of Large Sets of Idempotent Quasigroups. Eur. J. of Combin. 9, 83–89 (1988)
10. Zhang, H.: SATO: An Efficient Propositional Prover. In: Proc. of CADE, pp. 272–275 (1997)
11. Zhang, H., Stickel, M.: Implementing the Davis-Putnam Method. Journal of Automated Reasoning 24(1/2), 277–296 (2000)
12. Zhang, J., Zhang, H.: SEM: a System for Enumerating Models. In: Proc. of International Joint Conference on Artificial Intelligence, pp. 11–18 (1995)
13. Zhu, L.: Large Set Problems for Various Idempotent Quasigroups (July 2006)
14. Zhu, L.: Personal Communication (September 2007)