Searching for Doubly Self-orthogonal Latin Squares*

Runming Lu[1,2], Sheng Liu[1,2], and Jian Zhang[1]

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

Abstract. A Doubly Self Orthogonal Latin Square (DSOLS) is a Latin square which is orthogonal to its transpose to the diagonal and its transpose to the back diagonal. It is challenging to find a non-trivial DSOLS. For the orders $n = 2 \pmod 4$, the existence of DSOLS(n) is unknown except for $n = 2, 6$. We propose an efficient approach and data structure based on a set system and exact cover, with which we obtained a new result, i.e., the non-existence of DSOLS(10).

1 Introduction

Latin squares (quasigroups) are very interesting combinatorial objects. Some of them have special properties. It can be quite challenging to know, for which positive integer n, a latin square of size n (with certain properties) exists. Mathematicians have proposed several construction methods to generate bigger Latin squares from smaller ones. For the small Latin squares, some can be found or constructed by hand easily, but the others are very hard to generate. Computer search methods can be helpful here. In fact, many open cases in combinatorics have been solved by various programs [2,8].

In this paper, we study a special kind of Latin square named *doubly self-orthogonal Latin square* (DSOLS) which is related to the so-called perfect Latin squares [4]. In [1], Du and Zhu proved the existence of DSOLS(n) for $n = 0, 1, 3 \pmod 4$, except for $n = 3$. For the orders $n = 2 \pmod 4$, the existence of DSOLS(n) is unknown except for $n = 2, 6$. So the existence of DSOLS(10) is the smallest open case.

2 Preliminaries and Notations

A Latin square L of order n is an $n \times n$ table where each integer $0, 1, \ldots, n-1$ appears exactly once in each row and each column. We call each of the n^2 positions of the table a cell. For instance, the position at row x column y is called cell (x, y) and the value of cell (x, y) is denoted as $L(x, y)$, where $x, y, L(x, y)$ all take values from $[0, n-1]$. If $\forall i \in [0, n-1], L(i, i) = i$, L is called an idempotent Latin square.

Two Latin squares L_1 and L_2 are orthogonal if each pair of elements from the two squares occurs exactly once, or alternatively,

$$L_1(x_1, y_1) = L_1(x_2, y_2) \wedge L_2(x_1, y_1) = L_2(x_2, y_2) \rightarrow x_1 = x_2 \wedge y_1 = y_2$$

* This work is partially supported by the National Natural Science Foundation of China (NSFC) under grant No. 60673044. Corresponding author: Jian Zhang. We are grateful to the anonymous reviewers for their comments, to Lie Zhu and Feifei Ma for their help in this research.

J. Lee (Ed.): CP 2011, LNCS 6876, pp. 538–545, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Definition 1. *A DSOLS of order n, denoted as DSOLS(n), is a Latin square A which is orthogonal to both its transpose to the diagonal A^T and its transpose to the back diagonal A^*.*

A DSOLS A of order n can also be characterized using first order logic formulas:

$$A(x, y) = A(x, z) \rightarrow y = z \tag{1}$$

$$A(x, y) = A(z, y) \rightarrow x = z \tag{2}$$

$$A(x_1, y_1) = A(x_2, y_2) \wedge A(y_1, x_1) = A(y_2, x_2) \rightarrow x_1 = x_2 \wedge y_1 = y_2 \tag{3}$$

$$A(x_1, y_1) = A(x_2, y_2) \wedge A(n-1-y_1, n-1-x_1) = A(n-1-y_2, n-1-x_2)$$

$$\rightarrow x_1 = x_2 \wedge y_1 = y_2 \tag{4}$$

Table 1 gives an example of DSOLS(4).

Table 1. A DSOLS(4) and Its two Transposes

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

A

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

A^T

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

A^*

A closely related concept is *doubly diagonal orthogonal latin squares* (DDOLS) [3], which refers to a pair of orthogonal latin squares with the property that each square has distinct elements on the main diagonal as well as on the back diagonal. A DSOLS can be viewed as a special kind of a DDOLS. The existence problem for DDOLS has been solved completely later on.

3 Finding a DSOLS Using SAT and CSP

3.1 SAT Encoding of the Problem

We first encode the problem of finding DSOLS(n) into a SAT instance. Since each cell of Latin square can take one and only one value from the domain $[0, n-1]$, we introduce one boolean variable for each possible value of each cell. For each row $i \in [0, n-1]$, each column $j \in [0, n-1]$ and each candidate value $k \in [0, n-1]$, a boolean variable V_{ijk} is introduced. The variables V_{ijk} should satisfy some inherent constraints. For instance, each cell should take a value from $[0, n-1]$, so we have:

$$\forall i, j \in [0, n-1], V_{ij0} \vee V_{ij1} \vee \cdots \vee V_{ij(n-1)}$$

But each cell should not take more than one values from $[0, n-1]$ at the same time, so $\forall i, j \in [0, n-1]$, we have formulas like the following:

$$\neg V_{ij0} \vee \neg V_{ij1} \quad \neg V_{ij0} \vee \neg V_{ij2} \quad \cdots \quad \neg V_{ij(n-2)} \vee \neg V_{ij(n-1)}$$