The Troubles of Interior Design–A Complexity Analysis of the Game Heyawake

Markus Holzer and Oliver Ruepp

Institut für Informatik, Technische Universität München,
Boltzmannstrasse 3, D-85748 Garching bei München, Germany
{holzer,ruepp}@in.tum.de

Abstract. Heyawake is one of many recently popular Japanese pencil puzzles. We investigate the computational complexity of the problem of deciding whether a given puzzle instance has a solution or not. We show that Boolean gates can be emulated via Heyawake puzzles, and that it is possible to reduce the Boolean Satisfiability problem to Heyawake. It follows that the problem in question is NP-complete.

1 Introduction

Heyawake is one of many pencil puzzles published by the Japanese company *Nikoli Inc.* which specializes in logic games. Pencil puzzles have gained considerable popularity during recent years. The arguably most prominent example is the game of Number Place (jap. Sudoku), which first appeared as early as 1979 in an American magazine, but did not receive much attention until *Nikoli Inc.* published their version of the puzzle on the Japanese market. Being a big hit in Japan, the puzzle later became very popular around the whole world, and now the interest in other pencil puzzles is also rising.

As most other pencil puzzles, Heyawake (engl. "divided rooms") is played on a finite, two-dimensional rectangular grid. Compared to most other pencil puzzles however, Heyawake seems to be substantially more complicated due to its many rules. The grid is sub-divided into smaller rectangles (which are also called rooms, hence the name), and each of these rectangles may or may not contain a number. The sub-rectangles must form a disjoint partition of the whole grid. The goal of the game is to paint the cells of the board either white or black, according to the following rules:

1. Black cells are never horizontally or vertically adjacent.
2. All white cells must be interconnected. Diagonal connections do not count.
3. If a sub-rectangle contains a number, it must contain exactly that many black fields. Otherwise, any number of black cells is allowed.
4. Any horizontal or vertical straight line of white cells must not pass through more than 2 sub-rectangles.

Figure 1 shows an example Heyawake puzzle and its solution. The reader is encouraged to verify that the solution is unique. Lots of puzzles are available on

P. Crescenzi, G. Prencipe, and G. Pucci (Eds.): FUN 2007, LNCS 4475, pp. 198–212, 2007.

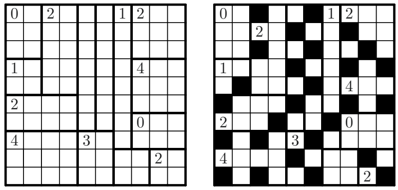

Fig. 1. Heyawake example puzzle (left) and its solution (right)

the internet, e.g., on `http://www.nikoli.com` or on `http://www.janko.at` (in German).

For a computer scientist, pencil puzzles are especially interesting from the computational complexity point of view. The probably most basic problem is finding a solution for a given puzzle, and in most cases, the corresponding decision problem ("is there a solution?") turns out to be NP-complete. Here NP denotes the class of problems solvable in polynomial time on a nondeterministic Turing machine. To our knowledge, the first result on pencil puzzles is due to Ueda and Nagao [8], who showed that Nonogram is NP-complete. Since then, a number of other pencil puzzles have been found to be NP-complete, e.g., Corral [2], Pearl [3], Spiral Galaxies [4], Nurikabe [6], Cross Sum (Jap. Kakkuro) [7], Slither Link [10], Number Place (Jap. Sudoku) [10], and Fillomino [10]. We contribute to this list by showing that Heyawake is NP-complete, too, proving the following theorem:

Theorem 1. *Solving a* Heyawake *puzzle is* NP*-complete.*

To this end, we show how to emulate Boolean circuits via Heyawake puzzles. We assume the reader to be familiar with the basics of complexity theory as contained in [5]. Hardness and completeness are always meant with respect to deterministic many-one log-space reducibilities.

2 Heyawake Is Intractable

To prove Theorem 1, we have to show that the problem in question is contained in NP, and that it is NP-hard. The containment in NP is immediate, since it is obvious that a nondeterministic Turing machine can guess a black and white pattern and check if that pattern constitutes a valid solution in polynomial time. It remains to prove the NP-hardness of the problem. We achieve this by showing how to reduce a 3SAT formula to Heyawake. We define the problem 3SAT as follows:

Instance: A finite set of Boolean variables $X = \{x_1, x_2, \ldots, x_n\}$ and a finite set of clauses $C = \{c_1, c_2, \ldots, c_m\}$, where each clause consists of 3 literals.

Question: If the input is interpreted in the obvious way as a 3CNF formula, is there an assignment for the variables such that the formula evaluates to true?