Exact Cover Problems: key

October 11, 2017

1 Regular Exact Cover problem

Use backtracking to solve the following exact cover problem, that is: **is there a set of rows containing exactly one 1 in each column?**

$$\begin{array}{c c} & \begin{array}{ccccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \end{array} \\ \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{array} & \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Initial configuration: in column 1, selecting row 1, as part of the solution.

$$\begin{array}{c c} & \begin{array}{ccccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \end{array} \\ \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{array} & \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Cover additional columns that are set in row 1 (i.e. 3 and 6), as well as rows 2, 3, 5, and 6 (conflict with row 1).
However, column 2 is now empty, although it has not been selected yet. BACKTRACKING: we uncover rows 6, 5, 3, and 2, as well as columns 7 and 4.

$$\begin{array}{c c} & \begin{array}{ccccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \end{array} \\ \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{array} & \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Selecting row 2 (as part of the solution).
Cover additional columns that are set in row 2 (i.e. column 4), as well as rows 1 and 4 (conflict with row 2).

$$\begin{array}{c c} & \begin{array}{ccccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \end{array} \\ \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{array} & \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

Recursing on the resulting matrix: cover row 4, as part of the solution.
Cover columns that are set in row 4 (i.e. 3, 5, 6), as well as row 5 (conflicts with row 4).

$$\begin{array}{cc} & \begin{array}{ccccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \end{array} \\ \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{array} & \left[\begin{array}{ccccccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}$$

Recursing on the resulting matrix. Cover row 6 (as part of the solution).
Cover columns 2 and 7, that are set in row 6.

$$\begin{array}{cc} & \begin{array}{ccccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \end{array} \\ \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{array} & \left[\begin{array}{ccccccc} 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array}$$

Recursing: the remaining matrix is empty. Success!

$$S = \{r_2, r_4, r_6\}$$

Note: an exact cover is best implemented by representing the matrix as a net of linked (vertical and horizontal) lists: rows and columns are covered by calling the DELETE(x) function on the corresponding data nodes, and backtracking to a former state of the matrix is easily accomplished through the RESTORE(x) function.

2 Generalized Exact Cover problem

Solve this generalized exact cover problem, that is: **Is there a set of rows containing *exactly one* 1 in each primary column** (c_1 through c_5), and ***at most one* 1 in each secondary column** (c_6 and c_7)?

We use the EXACTCOVER procedure but backtrack only if a *primary* column is a 0 column.

$$\begin{array}{cc} & \begin{array}{ccccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \end{array} \\ \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{array} & \left[\begin{array}{ccccccc} \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & 0 & 1 \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & 0 & 1 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & 1 & 1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 & 1 \end{array}\right] \end{array}$$

Initial configuration: in column 1, selecting row 1, as part of the solution.

$$\begin{array}{cc} & \begin{array}{ccccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \end{array} \\ \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{array} & \left[\begin{array}{ccccccc} \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & 0 & 1 \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & 0 & 1 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & 1 & 1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 & 1 \end{array}\right] \end{array}$$

Cover additional columns that are set in row 0 (i.e. 3 and 7), as well as rows 2, 3, 5, and 6 (conflict with row 1).
Column 6 is now a 0 column: even though it has not been selected yet, we are allowed to recurse, since column 5 is a secondary column.

$$\begin{array}{cc} & \begin{array}{ccccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 \end{array} \\ \begin{array}{c} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \\ r_6 \end{array} & \left[\begin{array}{ccccccc} \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & 0 & 1 \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & 0 & 1 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} & 1 & 1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 & 1 \end{array}\right] \end{array}$$

Select row 4. The resulting matrix is empty. Success!

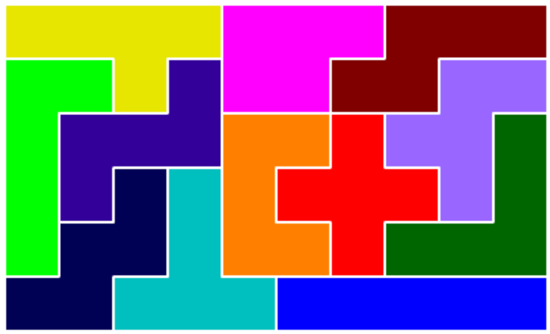
$$S = \{r_1, r_4\}$$

3 A standard exact cover problem: Pentomino

The Pentomino is a tiling problem involving

- 12 different tiles, each of them covering 5 cells
- a 6x10 grid

The following is one of the many solutions to the Pentomino problem:



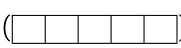
Design the exact cover matrix for the Pentomino problem:

- what should be represented by its columns?
- what should each row of the matrix represent?
- what is a solution to the problem?

Answer: A solution to the Pentomino problem meets the following constraints: (1) Each one of the 12 tiles have been used. (2) Each one the 60 cells in the grid are covered by exactly one tile. (3) No tile can be in 2 distinct positions.

From these constraints, we can derive for an exact cover matrix with 72 columns:

- each one of the first 12 columns represents a tile: column 1 for tile 1, column 2 for tile 2, ..., column 12 for tile 12.
- each one of the subsequent 60 columns represents a cell in the grid: column 13 for position (1,1), column 14 for position (1,2), ..., column 72 for position (6,10).

Each row in the matrix represents a way to insert a given tile on the grid. For instance, placing the "I"-tile () vertically in the top left corner is represented by a row where the following columns are set to 1:

- 1 (assuming that column 1 has been chosen to represent the "I"-tile)
- 13, 23, 33, 43, 53 -corresponding to positions (1,1), (2,1),(3,1),(4,1),(5,1) on the grid

A solution to the problem is a list of rows that selects exactly one 1 in each column of the matrix:

- this ensures that constraint (1) is met, since each one of the first 12 columns is eventually covered
- because exactly 1 row is selected for any given tile, constraint (3) is also met
- because exactly 1 row is selected for any given grid position, the solution assigns exactly 1 tile to this position, ensuring that constraint (2) is met

Sudoku puzzles are also instances of the standard exact cover problem. Designing the matrix suitable for solving sudokus is left as an exercise to the reader.