

Exact Cover Problems

October 8, 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?**

r ₀	1	0	0	1	0	0	1
r ₁	1	0	0	1	0	0	0
r ₂	0	0	0	1	1	0	1
r ₃	0	0	1	0	1	1	0
r ₄	0	1	1	0	0	1	1
r ₅	0	1	0	0	0	0	1

Initial configuration: in column 0, selecting row 0, as part of the solution.

r ₀	1	0	0	1	0	0	1
r ₁	1	0	0	1	0	0	0
r ₂	0	0	0	1	1	0	1
r ₃	0	0	1	0	1	1	0
r ₄	0	1	1	0	0	1	1
r ₅	0	1	0	0	0	0	1

Cover additional columns that are set in row 0 (i.e. 3 and 6), as well as rows 1, 2, 4, and 5 (conflict with row 0).

However, column 1 is now empty, although it has not been selected yet. BACKTRACKING: we uncover rows 5, 4, 2, and 1, as well as columns 6 and 3.

r ₀	1	0	0	1	0	0	1
r ₁	1	0	0	1	0	0	0
r ₂	0	0	0	1	1	0	1
r ₃	0	0	1	0	1	1	0
r ₄	0	1	1	0	0	1	1
r ₅	0	1	0	0	0	0	1

Selecting row 1 (as part of the solution).

Cover additional columns that are set in row 1 (i.e. column 3), as well as rows 0 and 3 (conflict with row 1).

r ₀	1	0	0	1	0	0	1
r ₁	1	0	0	1	0	0	0
r ₂	0	0	0	1	1	0	1
r ₃	0	0	1	0	1	1	0
r ₄	0	1	1	0	0	1	1
r ₅	0	1	0	0	0	0	1

Recurring on the resulting matrix: cover row 3, as part of the solution.

Cover columns that are set in row 3 (i.e. 2, 4, 5), as well as row 4 (conflicts with row 3).

r ₀	1	0	0	1	0	0	1
r ₁	1	0	0	1	0	0	0
r ₂	0	0	0	1	1	0	1
r ₃	0	0	1	0	1	1	0
r ₄	0	1	1	0	0	1	1
r ₅	0	1	0	0	0	0	1

Recurring on the resulting matrix. Cover row 5 (as part of the solution).

Cover columns 1 and 6, that are set in row 5.

r ₀												
r ₁	1	0	0	1	0	0	0	1	0	0	0	1
r ₂	0	0	0	1	1	0	1	0	1	0	1	0
r ₃	0	0	1	0	1	1	0	1	0	0	1	0
r ₄	0	1	1	0	0	1	1	0	1	1	0	0
r ₅	0	1	0	0	0	0	0	1	0	0	0	1

Recurring: the remaining matrix is empty. Success!

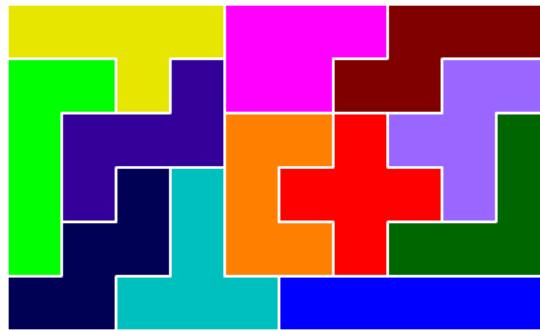
Note: an exact cover is best implemented by representing the matrix as a net of linked (vertical and horizontal) lists: rows and columns are covered by calling the `DELETE(x)` function on the corresponding data nodes, and backtracking to a former state of the matrix is easily accomplished through the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

2 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.

3 A generalized exact cover problem: 8-Queens

Design the exact cover matrix for the 8-Queen problem.

Answer: The 8-Queen problem is a generalization of the standard exact cover problem. The matrix has 2 sets of columns:

- The first 16 columns represent rank (columns 1 through 8) and file (columns 9 through 16) positions for a given queen
- The next 30 columns represent the up (first 15 columns) and down (next 15 columns) diagonals covered by a queen in the position coded by this row

Why is this a generalized exact cover problem? Each row in the matrix represents a way to place a single queen on the board. The solution set of rows has 8 elements, one for each queen. The first 16 columns are *primary* columns, to which the usual constraint applies: since the solution set of rows contains *exactly* one 1 for each of these columns, we are ensured that each queen is on its own row and column. The 30 diagonals columns are *secondary* columns: for those, the row selection constraint can be relaxed, so that *at most* one 1 can be selected for each column. This ensures that no two queens in different positions cover the same diagonal, without introducing the extra, undue requirement that each diagonal be covered by a queen: the algorithm might terminate with some diagonal columns still uncovered.