HLSudokuSolver Overview

When HLSudokuSolver launches, it loads a Sudoku puzzle from "www.websudoku.com". Below is a screenshot of a typical initial screen:

Rows (on)			**Given**	**Unsolved**	**Changed**	**Solved**		
Columns (on)								
Blocks (off)		Evil Puzzle 8,700,119,084			Unsolved Nodes: 123			
8	169	13569	19	359	4	359	7	235
35	2	3459	89	3589	7	34589	6	1
1357	1479	13459	6	2	59	34589	3489	358
4	19	1289	289	6789	3	15678	18	578
6	5	7	48	48	1	2	38	9
123	19	12389	5	6789	269	13678	138	4
257	467	2456	249	1	8	3479	2349	237
9	8	1246	3	46	26	147	5	27
125	3	1245	7	459	259	1489	12489	6

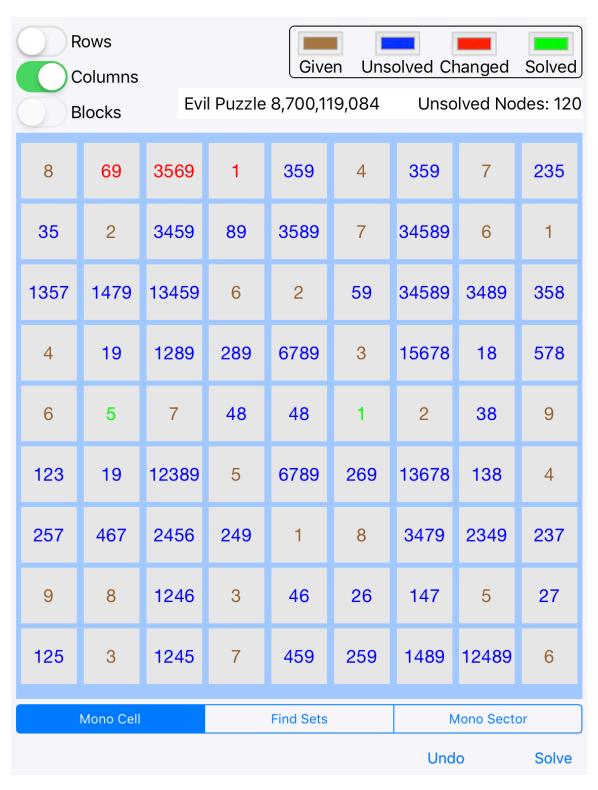| Mono Cell	Find Sets	Mono Sector

Undo   **Solve**

Note the 'Undo' button is dimmed. This is because we are at initial conditions and there is nothing to undo yet.

Notice on row 5 values '1' and '5' have been 'Solved' and are therefore green. Just from initial conditions these 2 values were solved by the pruning process that occurs after importing new data and after each time an algorithm runs on the data set.

The pruning process is run on rows, columns, and blocks. In the rows case, for each row, a list of solved cells are found and then those solved values are removed from all other cells in that row. Then this process is repeated for the columns and blocks.

Mono Cell Algorithm

The first solve operation to be described is 'Mono Cell' and it can be performed on rows and columns. It could also be applied to blocks but running the algorithm on rows and columns will find all there is to find using this approach. It checks on each row for an occurrence of a value being found in only one cell. If such cells are found, then those cells are marked as 'Solved'.

Rows

Columns

Blocks

Evil Puzzle 8,700,119,084   Unsolved Nodes: 120

8	69	3569	1	359	4	359	7	235
35	2	3459	89	3589	7	34589	6	1
1357	1479	13459	6	2	59	34589	3489	358
4	19	1289	289	6789	3	15678	18	578
6	5	7	48	48	1	2	38	9
123	19	12389	5	6789	269	13678	138	4
257	467	2456	249	1	8	3479	2349	237
9	8	1246	3	46	26	147	5	27
125	3	1245	7	459	259	1489	12489	6

Mono Cell	Find Sets	Mono Sector

Undo   Solve

Here on column 4, the value '1' was solved and after pruning row 1 columns 2 and 3 were pruned having the '1' removed from their data set.

The color red indicates that the values in that cell have changed during the last Solve operation. Typically you would perform this operation on both rows and columns at

the same time and then proceed to the next 'Solve' operation. Often, simply reapplying the same algorithm will continue to make progress. But at some point this may stop and another algorithm will need to be applied.
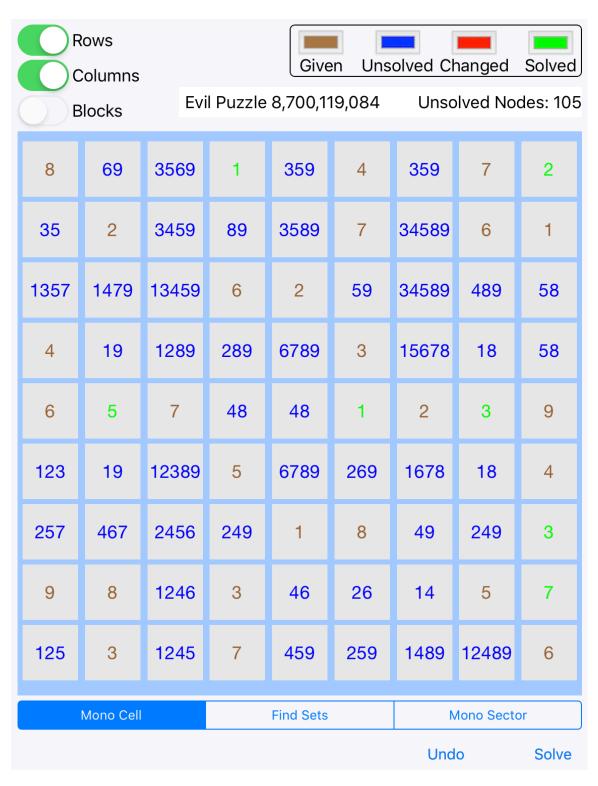
Find Sets Algorithm

Rows

Columns

Blocks

| Given | Unsolved | Changed | Solved |

Evil Puzzle 8,700,119,084   Unsolved Nodes: 110

8	6	359	1	359	4	359	7	235
35	2	3459	89	3589	7	34589	6	1
1357	47	13459	6	2	59	34589	49	358
4	19	1289	289	6789	3	15678	18	578
6	5	7	48	48	1	2	38	9
123	19	12389	5	6789	269	13678	138	4
257	47	2456	249	1	8	3479	249	237
9	8	1246	3	46	26	147	5	27
125	3	1245	7	459	259	1489	249	6

| Mono Cell | Find Sets | Mono Sector |

Undo   Solve

The 'Find Set' algorithm is by far the most powerful method. It can be applied to rows, columns, and blocks. Using columns as an example and the initial puzzle data we apply the Find Sets algorithm. On column 2 it finds 2 cells that contain the set {1, 9}. Those 2 cells contain the solutions for values '1' and '9'. It is either 1 then 9 or 9 then 1. Either way, values 1 and 9 can be removed from all other cells on that column. This solves for the value '6' and also allows for pruning in row 3 and 7. Further pruning occurs on column 3. The set {1,3, 8} is found in 3 cells in column 8 which allows for pruning in that column.

Mono Sector Algorithm

The 'Mono Sector' algorithm can be applied to rows and columns. To see how this works it is best to first run the Mono Cell on both rows and columns repeatedly until the Unsolved Nodes value doesn't change. Doing this on our initial data set gives us the puzzle below.

Evil Puzzle 8,700,119,084   Unsolved Nodes: 105

8	69	3569	1	359	4	359	7	2
35	2	3459	89	3589	7	34589	6	1
1357	1479	13459	6	2	59	34589	489	58
4	19	1289	289	6789	3	15678	18	58
6	5	7	48	48	1	2	3	9
123	19	12389	5	6789	269	1678	18	4
257	467	2456	249	1	8	49	249	3
9	8	1246	3	46	26	14	5	7
125	3	1245	7	459	259	1489	12489	6

| Mono Cell | Find Sets | Mono Sector |

Undo   Solve

Each row has 3 sectors. A row sector contains 3 consecutive columns that reside in the same block. For example, rows 1, 2, and 3 have 3 cells in block 1 (columns 1, 2, and 3), 3 cells in block 2 (columns 4, 5, and 6) and 3 cells in block 3 (columns 7, 8, and 9).

After applying Mono Sector to rows we have the result below.

Rows
Columns
Blocks

Evil Puzzle 8,700,119,084   Unsolved Nodes: 98

8	69	3569	1	359	4	359	7	2
35	2	3459	89	3589	7	3459	6	1
1357	1479	13459	6	2	59	34589	489	58
4	19	1289	29	679	3	15678	18	58
6	5	7	48	48	1	2	3	9
123	19	12389	5	679	269	1678	18	4
257	467	2456	249	1	8	49	249	3
9	8	124	3	46	26	14	5	7
12	3	124	7	459	259	1489	12489	6

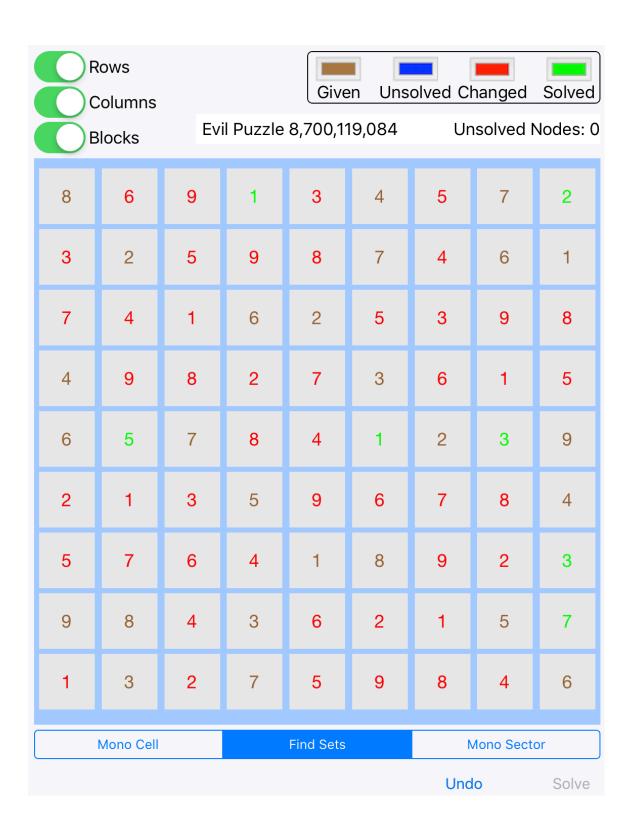| Mono Cell | Find Sets | Mono Sector |

Undo   Solve

Looking at row 2 column 7 we see that the value 8 has been removed. This was possible because the algorithm noticed that in block 2 the value 8 must be on row 2.

It must be on row 2 because in block 2 it's not on row 1 or 3. Therefore, it can't be in the first or last sector of row 2, as it must lie in the middle sector of row 2.

Let's look at the last row where the value '5' was removed from the first sector of row 9. This was possible because for block 8 the value 5 must be in row 9 (middle sector). Since we know that the value '5' must reside in either cell[9,5] or cell[9,6], we can prune 5 from cell[9,1] and cell[9,3].

Puzzle Solved

There are many different paths to solve most puzzles. Here is our puzzle solved with one application of Find Sets for rows, columns, and blocks using the data set from the previous solve operation (Mono Sector- rows).

Given	Unsolved	Changed	Solved	

Evil Puzzle 8,700,119,084   Unsolved Nodes: 0

8	6	9	1	3	4	5	7	2
3	2	5	9	8	7	4	6	1
7	4	1	6	2	5	3	9	8
4	9	8	2	7	3	6	1	5
6	5	7	8	4	1	2	3	9
2	1	3	5	9	6	7	8	4
5	7	6	4	1	8	9	2	3
9	8	4	3	6	2	1	5	7
1	3	2	7	5	9	8	4	6

Mono Cell	Find Sets	Mono Sector

Undo   Solve

Written By: Matthew Homer

Homer Labs

San Jose, CA

Last Modified: 12-23-2018