

Assignment 2: Encoding Sudoku into SMT

CS 6962 – Software Verification

February 25, 2015

Deadline: Wednesday, Mar 4, 2015 at 11:59pm MDT.

Sudoku Puzzle. Sudoku is a popular number-placement puzzle that originated in France in the end of the 19th century. Modern Sudoku was likely invented by Howard Garns from Connerville, Indiana and was first published in 1979 under the name “Number Place”. The objective of the puzzle is to place numbers 1 – 9 on a 9×9 grid, such that each number occurs only once in every row, every column, and every of the nine 3×3 sub-grids that compose the main grid. Sudoku puzzles are grids that have been partially occupied with numbers. The task is then to occupy the remaining fields in such a way that the constraints on rows, columns, and sub-grids are satisfied. A sample Sudoku problem and its solution are given in Figure 1. For more information about Sudoku refer to its WIKIPEDIA page at: <http://en.wikipedia.org/wiki/Sudoku>.

Your Task. Your task is to write a program that converts a Sudoku problem into an SMT formula such that a satisfying assignment for the generated formula (when appropriately translated) gives you a solution for the Sudoku problem.

Program input: A *file* (so the input of your program should be just a *filename*) containing a Sudoku problem given in the following form (each column is 3 characters wide to accommodate for numbers that have more than one digit; see bonus points below)

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

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

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

Figure 1: A sample Sudoku problem (left) and its solution (right).

Program output: A solution to the given Sudoku problem in the following form

```

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

```

Here are some additional notes:

- I recommend you use only the theory of linear integer arithmetic.
- I *strongly* recommend that you use Z3 as your SMT solver: <http://z3.codeplex.com/>. You can either invoke it as an external command from your program or use it as a library. You can assume that Z3 is in the PATH variable on my machine. If you prefer to use a different SMT solver, submit its binary with your solution.

Assignment Deliverables. Source code and a Linux (preferably) or Windows script/binary of your solution (sorry, no Mac OS). Also, a brief summary (at most one page, PDF format) explaining how your encoding works and how to invoke your solution. I want to be able to run your solution on my machine. Email me the deliverables.

Bonus Points. Parameterize your solution so that it works with different board sizes: 9×9 , 16×16 , 25×25 , 36×36 ,... Board size should be given as input to your solution.