

# Assignment 2: Encoding Sudoku into SMT

CS 6962 – Software Verification

September 6, 2012

**Deadline: Monday, Sep 24, 2012 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 Connorsville, 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  $9 \times 9$  grid, such that each number occurs only once in every row, every column, and every of the nine  $3 \times 3$  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

```
25..3.9.1
.1...4...
4.7...2.8
..52.....
....981..
.4...3...
...36..72
.7.....3
9.3...6.4
```

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

```
258736941
619824357
437915268
395271486
```

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).

```
762498135
841653729
184369572
576142893
923587614
```

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://research.microsoft.com/en-us/um/redmond/projects/z3>. 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.
- I *strongly* recommend that you implement your solution using Z3Py, which is a convenient Python API for Z3: <http://rise4fun.com/z3py/tutorial/guide>. Z3Py should make your life relatively easy. If you haven't used Python before, maybe it is time you give it a try :). Here is a short Z3Py script example:

```
from z3 import *

x = Int('x')
y = Int('y')
solve(x > 2, y < 10, x + 2*y == 7)
```

You can assume that the Z3 Python front-end directory will be in the PYTHONPATH environment variable on my machine.

**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.