Assignment 2: Encoding KenKen into SMT

CS 5100/6100 – Foundations of Computer Science

February 13, 2013

Deadline: **Wednesday, Feb 27, 2013 at 11:59pm MDT.**

KenKen Puzzle. KenKen is a math puzzle similar to Sudoku. A sample 4×4 KenKen puzzle and its solution are given in Figure 1. The basic rules are (from http://www.kenken.com):

1. Fill in a 4×4 grid with numbers from 1 to 4.
2. Each number occurs only once in every row and every column.
3. The numbers in each heavily outlined set of cells, called a *cage*, must combine in some order using the indicated mathematical operation to produce the preset target number in the top corner. For example, in Figure 1, number 16 in the top left corner is produced by multiplying 2, 1, 2, and 4.
4. Cages containing just one square should simply be filled with the target number.
5. A number can be repeated within a cage as long as it is not in the same row or column.

To learn more about KenKen and solve example puzzles visit:

- http://www.kenken.com
- http://en.wikipedia.org/wiki/KenKen

Your Task. Your task is to write a program that converts a 4×4 KenKen puzzle into an SMT formula such that a satisfying assignment for the generated formula (when appropriately translated) gives you a solution for the input KenKen puzzle.
Program input: A 4×4 KenKen puzzle given in the following form:

```
<number_of_cages_N>
<c1_target>,<c1_operation>,<c1_cell1>,<c1_cell2>,...
<c2_target>,<c2_operation>,<c2_cell1>,<c2_cell2>,...
...
<cN_target>,<cN_operation>,<cN_cell1>,<cN_cell2>,...
```

For example, the input describing the puzzle in Figure 1 is:

```
7
16,*,r1c1,r1c2,r2c2,r2c3
1,-,r1c3,r1c4
3,-,r2c1,r3c1
1,-,r2c4,r3c4
```

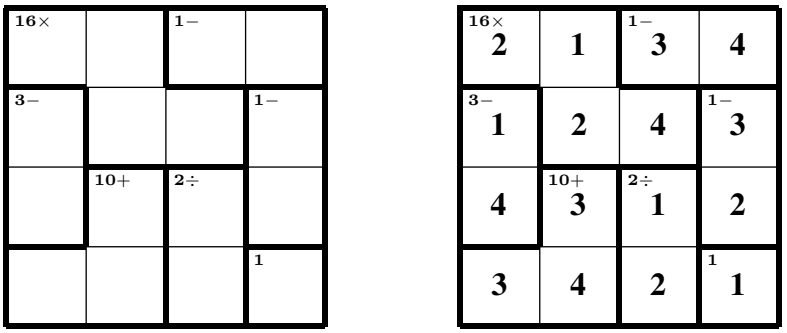

Figure 1: A sample KenKen puzzle (left) and its solution (right).

```
10,+,r3c2,r4c1,r4c2
2,/,r3c3,r4c3
1,=,r4c4
```

Program output: A solution to the input KenKen puzzle in the following form (see Figure 1, right):

```
2134
1243
4312
3421
```

Here are some additional notes:

- I recommend you use only the theory of 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.
- I *strongly* recommend that you implement your solution using Z3Py, which is a convenient Python API for Z3: http://rise4fun.com/z3py. Z3Py should make your life relatively easy. If you haven't used Python before, maybe it is time you give it a try :). You can assume that the Z3 Python front-end directory will be in the PYTHONPATH environment variable on my machine. Or you can code up your solution online using their convenient web-portal.

Assignment Deliverables. Source code and, if needed, a Linux (preferably) or Windows 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. Email me the deliverables.

Bonus Task (2 Points). Generalize your solution to support $N \times N$ KenKen puzzles.