Discrete Math – report

Team 9, 10 (Gyurim Kim, Yejin Kim, Jihee Park, Hanna Son, Susanna Jung, Yeon Hwangbo)

Index

1.intoduction

1-1 LIA

LIA is Linear Integer Arithmetic

1-2 SMT solver

Satisfiability Modulo Theories (SMT) problem is a decision problem for logical formulas with respect to combinations of background theories such as arithmetic, bit-vectors, arrays, and uninterpreted functions. Z3 is one of the SMT solver.

2.Puzzles

2-1 Sudoku*

2-1-1 Description

A. range constraints

Constraints: Sudoku puzzle has a 9 x 9 grid with 3 x 3 sub grids. Each cell has a number in 1 to 9. Each cell has a number in 1 to 9, and certain cells are assigned with one value in 1 to 9.

Define

solution

B. Row constraints

Constraints: Every row contains each of 1 to 9

solution

C. Column constraints

Constraints: Every column contains each of 1 to 9

solution

D. * constraints

Constraints: In Sudoku*, certain cells are marked with the * sign. At most 9 cells are marked with the * sign and no two cells marked with * have a same number

solution

E. sub grid constraints

Constraints: Every block (sub grid) contains each of 1 to 9.

solution

Test Example

```
? 2 ? 5 ? * ? 9 ?          6 2 1 5 7 8 3 9 4
8 ? ? 2 ? 3 ? ? 6          8 7 9 2 4 3 1 5 6
? 3 ? ? 6 ? * 7 ?          4 3 5 1 6 9 2 7 8
* ? ? ? * ? 6 ? ?          1 8 7 4 9 5 6 2 3
5 4 ? ? ? ? ? 1 9          5 4 3 7 2 6 8 1 9
? ? 2 ? ? ? 7 ? ?          9 6 2 3 8 1 7 4 5
? 9 * ? 3 ? ? 8 ?          7 9 4 6 3 2 5 8 1
2 ? ? 8 ? 4 ? * 7          2 5 6 8 1 4 9 3 7
? 1 ? 9 ? 7 ? 6 ?  ---------->  3 1 8 9 5 7 4 6 2
```

2-1-1 Discussion

Difficulty -

The star sudoku is based on original sudoku, so we made original sudoku at first. It was not hard to represent. However, it was a little difficult to derive this differently from all the star-marked values in the input. So, the way we came up with it was to put in the coordinates containing the stars and the value of n one by one, assuming that the sum should be one, because the value of the n should not overlap. From one to nine, each sum was declared one. And then, we could solve the problem. Also, we found another problem. We have to put pclose right after open pipe and make solution file. In addition, the class OSS is needed for understanding the system operating but several team members didn't take that class. So, there was more effort to let members understand it. We learn Z3 solver first by this assignment too, so if there is an error on that program, we need to search what it is and that gave us a slight inconvenience.

Lesson Learned –

We didn't have many opportunities to experience Linux, but through this assignment we could study and learn about Linux. We think we have to learn hard by studying Linux commands and how to make Makefile.

Idea –

When we were making star-Sudoku, we thought, "If Sudoku 9*9, except for small 3*3, what if there were no overlapping numbers in the diagonal?" But of course, input text file will be very important, and it is expected to be quite difficult.

Reference

http://theory.stanford.edu/~nikolaj/programmingz3.html#sec-intro

2-2

Fill-a-pix

2-2-1

-Description

Each puzzle consists of a grid containing clues in various places. The object is to paint the squares around each clue with black so that the number of painted squares, including the square with the clue, matches the value of the clue.

2-2-2

-Solution

Constraints of the solution:

Constraints of the solution:

1) In all element, the element should be 0 or 1

The P_{kij} is the solution of previous answer. The solution matrix save the answers from the 'solution file'.

2) But for multiple solutions, the current answer cannot be the same from the previous solutions. When the solution is K-th solution.

3) identify the starting clues - 0 and 9

-all nine squares - the one containing 9 - must be painted black

When the input is 9 in the matrix and the index i,j of input is in the range of 2 to M-1, or N-1, the surrounding element should be 1. When the index of input that is 9 is i and j,

-all nine squares - the one containing 0 - must be painted white

When the input is 0 in the matrix, the surrounding element should be 0. When the index of input that is 9 is i and j,

4) 6 along the edges

-all six squares in the highlighted area – the one containing the 6 and the five around it - must be painted black.

When the input is 6 in the matrix and the index i,j of input is side, the surrounding element should be 1. When the index of input that is 6 is i and j,

And if the i-1+k/3 and j-1+k%3 is outside the range of 1 to M and 1 to N, ignore the index.

```
for (k = 0; k < 9; k++){
    if (j+k%3 > N || i+k/3 > M || j+k%3 <= 0 || i+k/3 <= 0)
        continue;
```

5) 4 in a corner

-all four squares – the one containing the 4 and the three around it - must be painted black.

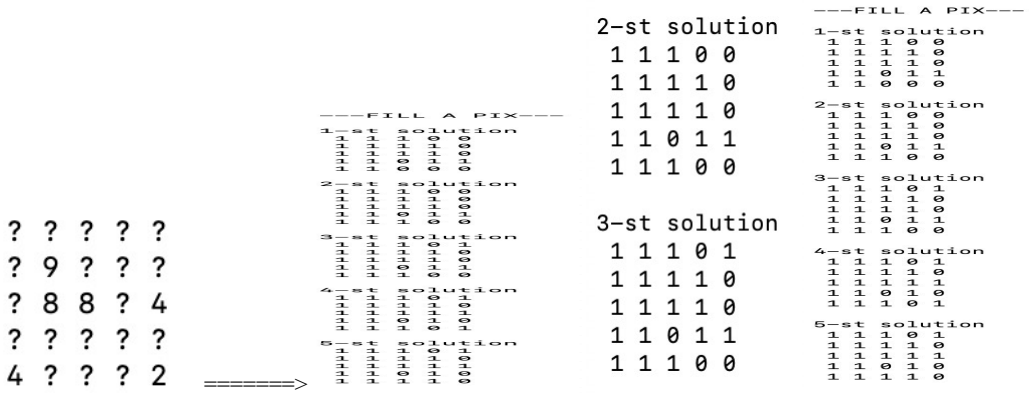
When the input in the matrix and the index i,j of input is corner, the surrounding element should be 1. When the index of input that is 4 is i and j,

And if the i-1+k/3 and j-1+k%3 is outside the range of 1 to M and 1 to N, ignore the index. The code is the same as above.

6) fill left over clues

Fill left over squares until the number of grid cell painted with black match with the clue number.

Test Example



-Discussion

Results –

When we read NxM grid with clues from the standard input where each line has initial settings of the cells of a row (- ? : no clue is given, - 1….9: a clue is given), we get a output grid with colorings where 1 represents black and 0 represents white.

It prints out “no solution” if there is no solution. If there are multiple solutions, it prints the solutions upto 5.

Difficulties-

After beginning with starting clues like 0,9 and 4 in corner and 6 in edge, it was difficult to think about the conditions to fill out left over cells with other clues.

Also, after making the program find a single solution, it was difficult to find multiple solutions.

Lessons Learned –

When we received our assignment, we were doubtful if we could solve it. However, as we worked together, we learnt the importance of teamwork and gained confidence. It was a good experience to learn about logics and how to use Z3 to check the satisfiability of logical formulas.

3-3 Numbrix

3-3-1 description

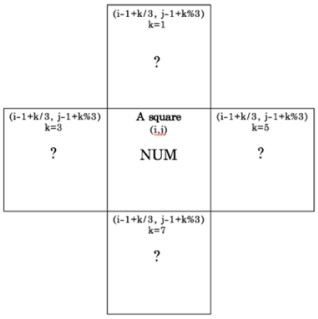
* constraints

(Assume that M is the number of rows, N is the number of columns, a_(i,j) is the value of the square located in i_th row, j_th column)

1) Each squares has a value from 1 to M*N.

2) The value of all squares has different value.

3) If any 'A square' has a NUM value, one of the squares adjacent to 'A square' must have a NUM+1 value. Except for cases where the NUM is M*N.



The currently completed code is the addition of only one condition to the basic condition that each square has a different value and the value of the square has one of the numbers of from 1 to N*M. This is the condition that only one of the squares adjacent to A square has a NUM+1 when the single square has a K value. However, it should be stated that this condition is not met for squares with N*M values because N*M is the end value. This condition applies to all squares, providing a single solution for each square to chain and eventually solve the numbrix.

3-3-2 discussion

The first code was divided into the case for squares in the corner, the case for borders except the corner, and the case for non-corner and case for non-corner. Each case is subdivided into when NUM is 1 and N*M, and other numbers. This will create inconvenience for us to fill out all nine conditions separately. Fortunately, We were able to find the current condition in which We had a meeting before completing all of the nine conditions and derive solution under a single condition. Based on this, I learned the importance of collaboration when programming and felt that I could achieve maximum efficiency by sharing ideas together rather than coding alone. Now, I think about another way to solve this puzzle, if we declare instead of and declare the k value in as 1,3,5,7 then we doesn’t have to use complicated formulas to indicate square locations.

3-3-3 test

1)

```
? ? ? ? ? ?              17 16 15 14  7  6
? ? 20 13 ? ?            18 19 20 13  8  5
? 26 ? ? 9 ?             27 26 21 12  9  4
? 25 ? ? 10 ?            28 25 22 11 10  3
? ? 23 36 ? ?            29 24 23 36 35  2
? ? ? ? ? ?   =======>   30 31 32 33 34  1
```

2)

```
1 2 3 4 7 8 9                 1  2  3  4  7  8  9
28 ? ? ? ? ? 10              28 27 26  5  6 11 10
29 ? ? ? ? ? 13              29 30 25 24 23 12 13
32 ? ? ? ? ? 14              32 31 36 37 22 21 14
33 ? ? ? ? ? 15              33 34 35 38 39 20 15
46 ? ? ? ? ? 16              46 45 44 43 40 19 16
47 48 49 42 41 18 17  =======>47 48 49 42 41 18 17
```

3)

```
1 24 ? ? ?
5 9 ? ? ?
? ? ? ? ?
2 ? ? ? ?
? ? ? ? ?
```

=======> no solution