

Automated Software Engineering (CSE 6323)
Homework 4 (Project description)
Student Name: Shafiul Azam Chowdhury, ID: 1001232942

Disclaimer: Some text is copied from Wikipedia/other online resources only for educational purpose.

Task 1: Test case generation for “n-puzzle” game using Z3

The **n-puzzle** is a [sliding puzzle](#) that consists of a frame of numbered square tiles in random order with one tile missing. If the size is 3×3 tiles, the puzzle is called the 8-puzzle or 9-puzzle, and if 4×4 tiles, the puzzle is called the 15-puzzle or 16-puzzle named, respectively, for the number of tiles and the number of spaces. The object of the puzzle is to place the tiles in order (Fig. 1) by making sliding moves that use the empty space.



Fig. 1: A solved 15-puzzle showing the goal/final state (Image: Wikipedia)

At any stage of the game, any number can be moved to a neighboring empty slot. Numbers can only be moved in left, right, upwards or downwards direction. For example, in Fig. 1 we can move the numbers 12 or 15 only to the empty slot.

In this task, I will focus on generating test case using Z3's Python API. More specifically, the task will be *finding the states from where we can reach the goal/final state using only n number of movements of the numbers.*

Task 2: Simple symbolic execution using Z3

Given a small program (C/Python code) I will be manually encoding its paths to use Z3's python API for symbolic execution of the program.

- **Task 2.1:** Checking for assertion violations in a small program. The paths will be manually identified and encoded and then be passed to Z3 (as input) for execution.
- **Task 2.2:** Using the code of Table 1 to:
 - Encode (as an input to Z3) the path corresponding to the "then" followed by the "else" followed by the "then", and the "else" followed by the "then" followed by the "else".
 - By querying Z3, check whether the two paths have satisfiable constraints? To which (a,b,c) triplets do they correspond to?

```
#define LENGTH 4
#define PRIZE 2

void check(int a, int b, int c){
    int position= 0;

    if(a==1){
        position=(position + LENGTH - 2) % LENGTH;
    }else{
        position=(position + 1) % LENGTH;
    }

    if(b==1){
        position=(position + 1) % LENGTH;
    }else{
        position=(position + LENGTH - 2) % LENGTH;
    }

    if(c==1){
        position=(position + LENGTH - 2) % LENGTH;
    }else{
        position=(position + 1) % LENGTH;
    }

    if(position==PRIZE){
        printf("you win :) \n");
        return;
    }
    printf("you lose :( \n");
    return;
}
```

Table 1

This task idea is mainly collected from: <https://www.ida.liu.se/~TDDC90/labs/LAB2-STATIC-2015.pdf>