

# PAVT Assignment 1

Akash Shivaji Varude

Roll No: 231110006

## 1) Deliverables Location:

All test files (.tl extension files) of my submission are stored in Tests folder in kachuacore folder.

Fuzzsubmission.py file (in which I've written mutation, compare Coverage, Update Coverage code) is present in submissions folder

## 2) Instructions to run fuzzer on **my created** test.tl files (.tl extension files) :

- I have created 5 test files. These are present in 'Test' folder under 'kachuaCore' folder
- Please use only two parameters :x and :y in all of my test.tl files
- I have used parameter values in kachua commands (e.g. **forward :x/2**  
**left :y/3** etc)

hence to make sure kachua is running in visible whiteboard please use :x and :y values within range (-500 to 500)

However if you want to use my mutation on your test files(.tl files) you can use any number of parameters and any value (not limited to specific range) you want.

## Limitations about parameters:

1. Parameters's values can be anything however, I've written test.tl files that runs on parameter's (:x and :y) values in the range **-500 to 500** only. To make kachua running in visible whiteboard.

## Explanation about Mutate function:

- I've randomly chosen number of parameters to mutate also which parameter to choose is also random.
- I have used **five types of mutations** and which mutation to choose is also random
- **First Mutation:** Addition or subtraction of random number from parameter value. Also deciding whether to choose addition or subtraction is also random.
- **Second Mutation:** Changing sign of parameter value
- **Third Mutation:** Changing parameter value to some random value
- **Fourth Mutation:** Setting Parameter value to 0
- **Fifth Mutation:** Performing bitwise XOR with bitmask of length 8. Bitmask is a binary value between 0 to 0b11111111. This value is chosen randomly

## Sample Outputs when running fuzzer for (60 seconds) on :

### 1. test1.tl :

```
C:\Windows\System32\cmd.exe
Coverage : [0, 1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 17, 18, 19, 20, 21, 22, 36, 37, 38, 39, 40, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 67, 71, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 104, 105, 106, 107, 108],
Corpus:
Input 0 : {'x': -175, 'y': 250}
Input 1 : {'x': -175, 'y': 270.2329473684438}
Input 2 : {'x': 175, 'y': 247.86393724590457}
Input 3 : {'x': 0, 'y': 132.21181749797475}
Input 4 : {'x': 175, 'y': -247.86393724590457}
Input 5 : {'x': 137.1183338449835, 'y': 0}
Input 6 : {'x': 175, 'y': 14.403443113706714}
Input 7 : {'x': 137.1183338449835, 'y': 87}

D:\Mtech IITK\PAVT Coding here\Chiron-Framework\KachuaCore>
```

### 2. test2.tl :

```
C:\Windows\System32\cmd.exe
Coverage : [0, 1, 2, 23, 24, 25, 28, 29, 30, 31, 32, 33, 37, 38, 39, 40, 41, 60, 61, 62, 67, 68, 69, 70, 71, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 87, 88, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 105, 106, 107, 108, 109, 110, 111, 112, 113, 115, 116, 117, 118, 119, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 145, 147, 148, 149, 150],
Corpus:
Input 0 : {'x': 0, 'y': 0}
Input 1 : {'x': 202, 'y': 249}
Input 2 : {'x': 0, 'y': -0.007214348509852735}
Input 3 : {'x': -202, 'y': 249}
Input 4 : {'x': 197, 'y': 0}
Input 5 : {'x': -197, 'y': 0}
Input 6 : {'x': 0.42541269394753195, 'y': 0}
Input 7 : {'x': -202, 'y': 41}
Input 8 : {'x': 0.42541269394753195, 'y': 235.61052510944762}

D:\Mtech IITK\PAVT Coding here\Chiron-Framework\KachuaCore>
```

### 3. test3.tl :

```
C:\Windows\System32\cmd.exe
Coverage : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 48, 49, 50, 62, 63, 64, 65, 66, 77, 78, 79, 82, 83, 84, 87, 88, 89, 90, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 106, 107, 108, 109, 110, 111, 112, 113],
Corpus:
Input 0 : {'x': 0, 'y': 0}
Input 1 : {'x': -0.058384275901146955, 'y': 138}
Input 2 : {'x': 87, 'y': -71.16482856273151}
Input 3 : {'x': 0, 'y': 231}
Input 4 : {'x': 0.5181631380628731, 'y': 138}
Input 5 : {'x': 87, 'y': 0}
Input 6 : {'x': 443.5581644325275, 'y': 259.61444741508456}
Input 7 : {'x': 443.5581644325275, 'y': 500}
Input 8 : {'x': 442.77380955198356, 'y': -259.61444741508456}

D:\Mtech IITK\PAVT Coding here\Chiron-Framework\KachuaCore>
```

#### 4. test4.tl :

```
C:\Windows\System32\cmd.exe
Coverage : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79],
Corpus:
Input 0 : {'x': -500, 'y': -100}
Input 1 : {'x': -399.6352160875241, 'y': -100.60180503149638}
Input 2 : {'x': -441.7674968478473, 'y': 0}
Input 3 : {'x': 399.6352160875241, 'y': -100.60180503149638}
Input 4 : {'x': 399.6352160875241, 'y': 110.63949081769437}
Input 5 : {'x': -399.6352160875241, 'y': 100.60180503149638}
Input 6 : {'x': -399.6352160875241, 'y': 233.907889409689}

D:\Mtech IITK\PAVT Coding here\Chiron-Framework\KachuaCore>
```

#### 5. test5.tl :

```
C:\Windows\System32\cmd.exe
Coverage : [0, 1, 2, 3, 4, 5, 6, 7, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72],
Corpus:
Input 0 : {'x': 30, 'y': -50}
Input 1 : {'x': 30.48503473546089, 'y': -60.83491002488447}
Input 2 : {'x': -44.53720449979862, 'y': -50}
Input 3 : {'x': -233, 'y': -50}
Input 4 : {'x': 30, 'y': 50}
Input 5 : {'x': -30.48503473546089, 'y': 60.83491002488447}

D:\Mtech IITK\PAVT Coding here\Chiron-Framework\KachuaCore>
```

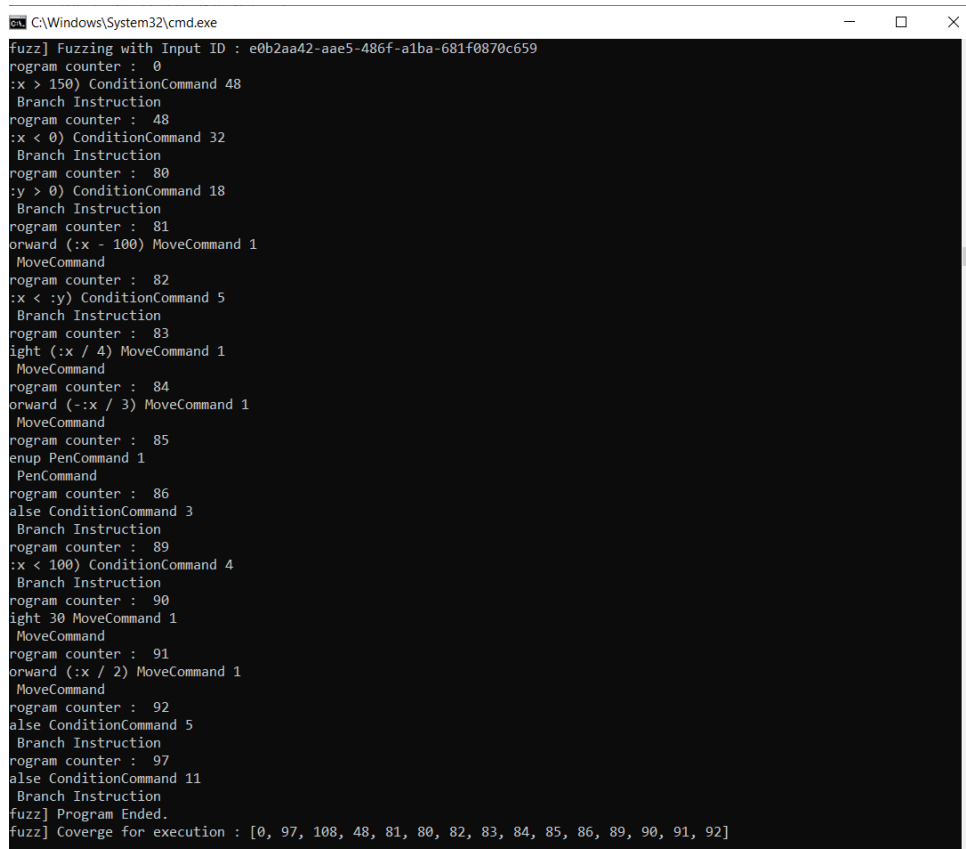
## Bug Report:

Occurs when fuzzer is tested on test1.tl file

No matter what the coverage is, **every time**(for given seed value and also for every mutated value) **it includes '108'** in coverage metrice. Below is screenshot of coverage for one of the mutated input.

Expected Coverage: [0, 97, 48, 81, 80, 82, 83, 84, 85, 86, 89, 90, 91, 92]

Actual Coverage [0, 97, 108, 48, 81, 80, 82, 83, 84, 85, 86, 89, 90, 91, 92]

A screenshot of a Windows command prompt window titled "C:\Windows\System32\cmd.exe". The window displays the output of a fuzzer test. The output shows a series of program counter values and instructions, including branch instructions and move commands. The final line of the output is "fuzz] Coverage for execution : [0, 97, 108, 48, 81, 80, 82, 83, 84, 85, 86, 89, 90, 91, 92]". The window has a standard Windows title bar with minimize, maximize, and close buttons.

```
fuzz] Fuzzing with Input ID : e0b2aa42-aae5-486f-a1ba-681f0870c659
rogram counter : 0
:x > 150) ConditionCommand 48
  Branch Instruction
rogram counter : 48
:x < 0) ConditionCommand 32
  Branch Instruction
rogram counter : 80
:y > 0) ConditionCommand 18
  Branch Instruction
rogram counter : 81
orward (:x - 100) MoveCommand 1
  MoveCommand
rogram counter : 82
:x < :y) ConditionCommand 5
  Branch Instruction
rogram counter : 83
ight (:x / 4) MoveCommand 1
  MoveCommand
rogram counter : 84
orward (-:x / 3) MoveCommand 1
  MoveCommand
rogram counter : 85
enup PenCommand 1
  PenCommand
rogram counter : 86
alse ConditionCommand 3
  Branch Instruction
rogram counter : 89
:x < 100) ConditionCommand 4
  Branch Instruction
rogram counter : 90
ight 30 MoveCommand 1
  MoveCommand
rogram counter : 91
orward (:x / 2) MoveCommand 1
  MoveCommand
rogram counter : 92
alse ConditionCommand 5
  Branch Instruction
rogram counter : 97
alse ConditionCommand 11
  Branch Instruction
fuzz] Program Ended.
fuzz] Coverage for execution : [0, 97, 108, 48, 81, 80, 82, 83, 84, 85, 86, 89, 90, 91, 92]
```