Assignment #4: Abstract Interpretation

Name: Yashvir Singh Nathawat

Roll No: 231110059

Mtech CSE

Email: ysnathawat23@iitk.ac.in

CS639: Program Analysis Verification and Testing

IMPLEMENTATION:

The primary objective of this assignment is to monitor kachua's movements during the program and determine the final potential range of the coordinates where Kachua rests. Additionally, there is a region known as "Magarmach's," which is defined as two sets of "x" and "y" coordinates that combine to form a rectangle. The primary goal is to confirm that the kachua never sleeps in the magarmach's area.

1) **Idea of Abstract Interpretation**
   Thus, "Abstract Interpretation" notions are used to accomplish this. Concrete values in the form of sets are essentially transformed into an abstract interval domain for abstract interpretation using the interval domain, where each set is represented by a range of values. This range extends from the set's lowest value to its largest value.

2) **How we track safe region**

We track all the coordinates x and y also all variables and track these value for the whole program.

3) **Abstraction Interpretation is done on:**

- The x-coordinate position of kachua
- The y-coordinate position of kachua
- The direction of kachua [‘l’, ’r’, ‘d’, ‘t’] Since, it is mentioned that the rotations can only be in multiples of 90 degrees, the direction of kachua at any point of time can be any one amongst the direction of 4-axes
- Variables from ‘a’ to ‘z’

4) Initialization:

First, we must initialize the single statement basic block's "In" of the start node. Thus, we initialize the x- and y-ranges with [0,0] to accomplish this. In addition, the kachua faces in the conventional '+r' direction at first. Moreover, initial values of [-300,300] for variables are assumed.

5) Transfer Function:

It calculates the ‘OUT’ of blocks using the ‘in’ values. I have implemented transfer function for following operations:

- Assigment Command
- Forward
- Backward
- Left
- Right
- Conditional Operators

6) **Meet Function:**

One or more "outs" of basic blocks are incoming at "ins" of a specific basic block in the meet function. To obtain the in of the current basic block, we must therefore combine these numerous outs of earlier basic blocks. Therefore, in the meet function, we take the range of x-coordinate to be [min{a,c}, max{c,d}] if the x-coordinate range of the previous outs is [a,b] and [c,d]. Additionally, this meet can only occur if all incoming and outgoing values have the same direction. However, as we approach the bottom of the direction lattice, we are unable to make any statements if the directions of our outs differ.

Assumptions:

The provided code has several assumptions:

1. Initial conditions for all variables are assumed to be [-300,300]
2. Initial condition for x and y coordinate is [0,0] and direction assumed is ‘r’
3. Condition commands are of format :a >= 20 and wont work if 20>=:a is used.
4. Variables can be only [‘a’,’b’,’c’,’d’,’e’]

Limitations:

The provided code has several limitations:

1. I haven't yet dealt with cases involving repeated statements in the Kachua software because handling these scenarios involved a great deal of complexity. Therefore, given that the safest way to solve this issue is to state that the "Kachua is unsafe," which means that if the kachua gets

into the Magarmach region is a secure option because we're not being inaccurate in this instance for any instances where the kachua truly landed in risky area, but we argue that it doesn't.

Therefore, whenever I come across repeated statements in the program I claim that the magarmach's territory is where the kachua lands. This remedy It may not be exact, but it is unquestionably secure.

2. Variables are only 5 which are ‘a’ to ‘z’.
3. Case when there are arithmetic expressions in expression is also not handled in my implementation.