**Assignment 5:Program Synthesis Using Symbolic Execution**

This Assignment was about implementing the a function that will basically take as argument the parsed command and the IR of a program which do not have any constants on some input variable then it intake a json file which contain the symbolic execution result of a program which had some variable as input variables and some as constants and now the task was to come with some concrete values for the constants which will make both the programs equivalent semantically.

Now, to do this basically we need to match paths followed for certain values of inputs in both the programs and then we shall equate them using an smt solver to get the result out.

There are two of ways of doing this which I found:

1. First perform symbolic execution on the program which has constant variables using this result for each inputs of each path execute the second program and finally equate the results variable wise using a smt solver to get the final result.Then,also inorder to be sure that the second program does’nt have any unvisited paths left perform symbolic execution on that too and take input values for each path and match constraints with the symbolic execution result of program 1’s if any path constraint is satisfied then equate the corresponding variables there too to be sure.
2. Second way is do not execute the second program as above rather perform the constraint matching and equate as above just it in both direction.First take program 1 and compare constraints for second one if any path constraints matches then equate variable encodings.Similarly perform this for 2nd program also.

For my assignment I have followed the First approach.

**Limitations/Assumptions:**

* I have assumed here that both the programs will have same number of input variables.
* The second program will no constant variables.
* This code is not tested for repeat statements.So can’t say what can go wrong.
* It is dependent code on Symbolic execution hence if we have any error or any missing or bug in that then this will also might crack.

**How to run ?**

1. First go to KachuaCore folder.
2. Execute the symbolic execution on the first program.
3. Execute the optimize program on second program (to generate the .kw file).e.g-“ ./kachua.py -O example/test0.tl”
4. The above will generate an Optimized.kw file using it run the main execution.e.g-“  python3 symbSubmission.py -b Optimized.kw -e '["x", "y"]'”
5. The result that is the various values for the constant variables will be printed on console.