**What is Symbolic Execution?**

Symbolic execution is a software testing technique that is useful to aid the generation of test data and in proving the program quality.

**Steps to use Symbolic Execution:**

- The execution requires a selection of paths that are exercised by a set of data values. A program, which is executed using actual data, results in the output of a series of values.

- In symbolic execution, the data is replaced by symbolic values with set of expressions, one expression per output variable.

- The common approach for symbolic execution is to perform an analysis of the program, resulting in the creation of a flow graph.

- The flowgraph identifies the decision points and the assignments associated with each flow. By traversing the flow graph from an entry point, a list of assignment statements and branch predicates is produced.

**Issues with Symbolic Execution:**

- Symbolic execution cannot proceed if the number of iterations in the loop is known.

- The second issue is the invocation of any out-of-line code or module calls.

- Symbolic execution cannot be used with arrays.

- The symbolic execution cannot identify of infeasible paths.

**Symbolic Execution Application:**

- Path domain checking

- Test Data generation

- Partition analysis

- Symbolic debugging