This paper covers the basic problems of testing finite state machines and briefly presents some general principles and methods for doing the same. The author presents a background of Finite state machines and gives a formal definition of an FSM in the second section. A Finite state machine is a model used to design sequential circuits, computer programs and communication protocols. He also talks about initial state uncertainty and current state uncertainty in which there can be more than one possible start state for a given sequence of inputs generating a fixed output. Successors tree have been explained next, which shows the behavior of the FSM for all possible inputs, starting from the initial.

Tests are done by applying a sequence of inputs and observing the output. There are five fundamental problems in testing finite state machines. First is to determine the final state after the test, second is to identify the unknown initial state, third is to verify if the machine is in the initial state or not, forth is conformance testing i.e. to verify if the machine behaves as an equivalent specification machine or not and last is to identify an unknown machine. In this paper, the author focuses on two main questions, whether there is a test sequence that solves the problem and if there is, how hard it is to find this sequence. Problem 1 can be addressed through homing and synchronizing sequence. Homing sequence is used to determine the final state of machine through a given input sequence. Synchronization sequence makes the machine to transition into a specific final state.

The next two problems of state identification and verification are addressed in section 3. Distinguishing sequence is used to solve the problem of state identification. There are two types of distinguishing sequence, preset and adaptive. In preset, the sequence is fixed ahead of time and the output sequence for is different for each initial state. In adaptive, the next input depends on the current output and it is a decision tree. To build an adaptive sequence, a two part algorithm is explained in which, first a splitting tree is generated where every node is a set of states and the root contains all the states. In the second part, this tree is used to generate the adaptive sequence. There are some machines that do not have a distinguishing sequence.

For state verification, UIO sequence is used. UIO sequence of a state S is an input sequence such that the machine would particular output sequence iff it was in a particular state. There are no efficient algorithms for finding UIO sequence. However, if an FSM has distinguish sequence then every state has an UIO sequence which is the input from the root of the decision tree to that particular state.

The next section is on Conformance Testing. In conformance testing, a machine A is given as a black box, and by observing its Input/Output behavior, it is to be determined whether this machine is a proper implementation of a complete specification or not which is provided as an FSM B. Certain assumptions are made for making this test to be possible. A is assumed to be strongly connected and a reduced FSM. Implementation of B does not change during the test and B has no more states than A. To do conformance testing, a homing sequence is applied to B to bring it to an initial state. Then a checking experiment is run to verify if B is isomorphic to A. A checking sequence is an input sequence such that it distinguishes A from all other machines with the same number of states, starting from a given initial state. Machines those are not isomorphic to A, produce different outputs. There is a lot that goes into conformance testing, a few of which have been covered in details in this section.

In the conclusion, the author mentions that it is still not known how to construct checking sequences deterministically in polynomial time. A table has been given to summarize the problems and compare the test sequences and their complexity.