Compilation and Testing:

The program is compiled using Visual Studio 2012 , an attached exe is found also on the folder.

For compiling , just open the project using Visual Studio.

Upon launching the app , you ‘ll need to enter relative / full path of the mfsm.

Afterwards , for each machine specified in the mfsm , you ‘ll need to enter a sequence of inputs representing the transitions within each declared fsm.

In case there are nested fsms , inline input on each state is mandatory.

The program , after running on each fsm a different thread , it will store the output of each fsm on a different string stream which would eventually guarantee isolation of the output even though if context switching occurs within printing out to the screen . By using that trick , you would avoid using locks to print out consistently to the screen

Those stringstreams are later transformed to strings and outputted simply to the screen.

By the way , locks are essentially important while the add instruction and out instruction.

Below are runtime snapshots showing the program output while running samples

The samples can be found in the sample dir . On each element , of the machine section , it should contain a relative / absolute path.

Class Diagrams are automatically generated using Visual Studio



