Finite State Machines

Tutorial

Dr S. S. Chandra

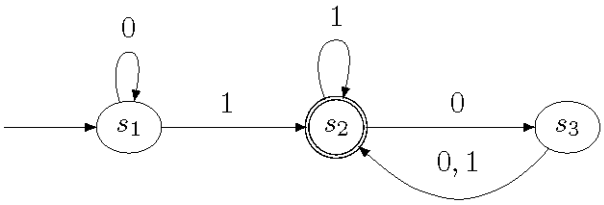
Finite state machines (FSMs) are simple computational constructs that have no read/write memory allowing the creation of simple devices. It is capable of accepting or rejecting a string of certain inputs to complete a simple computation.

In this tutorial, you will solve fundamental FSM related problems and explore concepts such as transition diagrams/tables, as well as properties of regular languages associated with FSMs.

Problem 1

Define the main components of an FSM and their use during a computation. Use these definitions to describe how a basic automatic door with pressure plates sensing users can be represented using FSMs. Be sure to include all the components necessary, critical elements of an FSM and assumptions of your system in your answer.

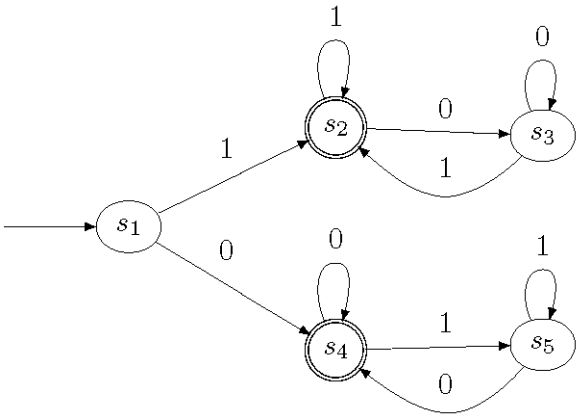
Problem 2



For the given FSM above

a) List the set of all possible states.
b) Write the transition table.
c) Show that it accepts the strings $w = 10100$ and $u = 0001$.
d) What type of strings does this machine accept?

Problem 3



For the given FSM above

a) List the set of all possible states.
b) Write the transition table.
c) Show that it accepts the strings $w = 101$ and $u = 00$.
d) What type of strings does this machine accept?

Problem 4

Design and define the following FSMs using either a transition table or state diagram

a) An FSM M_1 that only accepts set of strings containing even zeros
b) An FSM M_2 that only accepts set of strings containing odd ones
c) An FSM M_3 whose language is the union of the languages of the two machines. In other words, if the language accepted by M_1 is A and language accepted by M_2 is B then construct M_3 such that $A \cup B$, where the alphabet for $M_3$is also the same as alphabet for $M_1$and M_2.

Problem 5

Describe the main advantages and disadvantages of an FSM in terms of computation within roughly 250 words. Be sure to include at least one example and discussions on the consequence of having only a fixed number of finite states.

Problem 6

Design a simple non-playable character (NPC) or an animal's artificial intelligence (AI) system based on FSMs (without any code) for a game in a genre from the list in the appendix. For example, an AI system for a (non-interactable) mouse living in a dungeon part of an RPG game. The desired FSM AI system is to (at least) be able to handle the NPC's/animal's interaction after seeing a player and/or approaching it, as well as when the player is not detected or around. Ensure to include the transition diagram, assumptions and other important elements necessary for your AI system.

Appendix

Game genres allowed:

- RPG
- Sports
- FPS
- RTS/MOBA

If you would like another genre, please seek approval from the teaching team before designing your system.