# Finite state machines

A FSM It is an abstraction of an actual device. It shows the state that it is in – this is one state of a finite number of states. The state can transition to another state depending on the input.

## Please complete the notation table below

|  |  |
| --- | --- |
| **Symbol** | **Meaning** |
|  | State |
|  | Start state |
|  | End state |
|  | Transition |

## State transition diagram and State transition table

A ticket machine in a car park requires two inputs: money to be put in and the green button to be pressed. The double circle is the state in which a ticket is issued.

Input: Button pressed

Input: Money inserted

In this case:

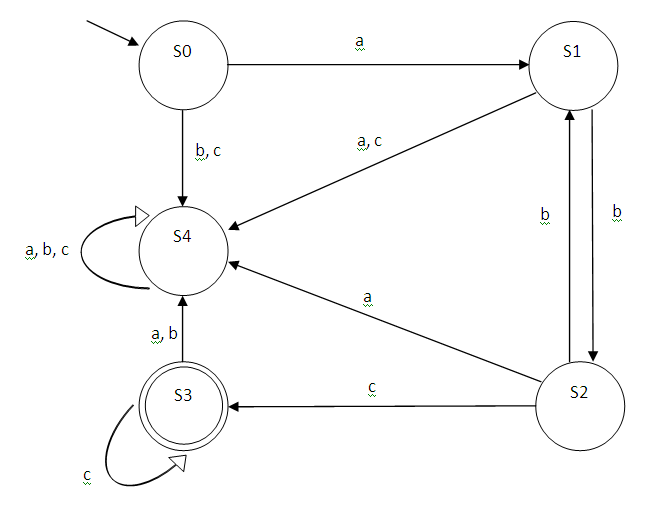
* S0 is the machine in its idle state, waiting for an input
* S1 is its state after the \_\_\_\_\_\_\_\_\_\_ has been put in
* \_\_ is its state after the \_\_\_\_\_\_\_\_\_\_ has been pressed. This is the accepting state

### Please complete the state transition table below

|  |  |  |
| --- | --- | --- |
| Input | Current state | Next state |
|  |  | S1 |
| \_\_\_\_\_\_\_\_ pressed | S1 |  |

## State transition diagram

This FSM below is used to check that the rules of programming language are being followed. It is a simplified example using just the letters a, b and c, though in real life the FSM could be set up to represent all of the acceptable words and combinations of words usable in any particular programming language.

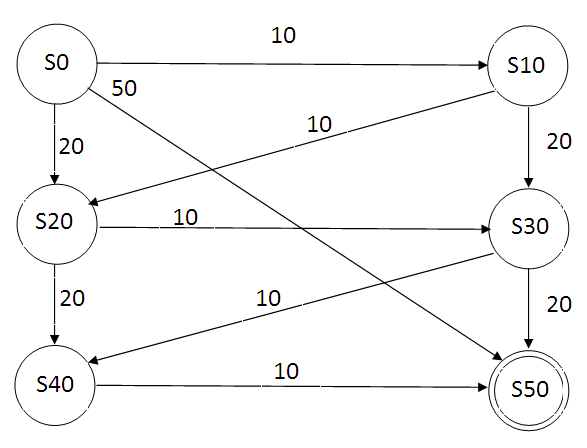


Looking at the diagram you can see whether certain combinations of letters are acceptable or not, for example:

* abc is an acceptable combination True / False
* abcc is an acceptable combination True / False
* acb is an acceptable combination True / False
* abca is an acceptable combination True / False

|  |  |  |
| --- | --- | --- |
| Input | Current state | Next state |
| a | S0 | S1 |
| b |  | S2 |
|  | S1 |  |
| a | S1 | S4 |
| b |  | S1 |
| c | S2 |  |
|  | S2 | S4 |
| b | S0 | S4 |
| c |  | S4 |
| a | S3 |  |
|  | S3 | S4 |
| c |  | S3 |
|  | S4 | S4 |
| b | S4 |  |
|  | S4 | S4 |

The following FSM represents a saving machine that accepts £10, £20 and £50 notes. Once the total of £50 is entered the user is awarded a token.



1. What is the accepting state?

1. What are inputs and corresponding new states from S0?

|  |  |
| --- | --- |
| Input | New state |
| 10 |  |
|  |  |
|  |  |

1. Apart from 10, 20, 20, list two inputs that would result in the accepting state?

**10,20,20**

1. Write out a state transition table for the diagram

|  |  |  |
| --- | --- | --- |
| Input | Current State | New State |
| 10 | S0 | S10 |
| 20 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |