

# Finite automata



**ASSIGN  
BUSTER**

The symbols of the sequence are presented sequentially to a machine  $M$ .  $M$  responds with a binary signal to each input. If the string scanned so far is accepted, then the light goes on, else the light is off. A language acceptor \*

Lesson 3 employs the treatment of this subject as found in *Machines, Languages, and Computation* by Denning, Dennis and Qualitz, Prentice-Hall. Transducer Abstract machines that operate as transducers are of interest in connection with the translation of languages.

The following transducer produces a sentence  $(l_1 l_2 \dots l_n)$  in response to the input sentence  $s(1) s(2) \dots s(m)$  translated into a specific sentence of an output language. Generator When  $M$  is started from its initial state, it emits a sequence of symbols  $(1) r(2) r(i) r(t)$  from a set known as its output alphabet. We will begin our study with the transducer model of abstract machine (or automaton). We often refer to such a device as a Finite State Machine (FSM) or as an automaton with output.

**Finite State Machine (FSM)** The FSM model arises naturally from physical settings in which information-denoting Only a finite number of operations may be performed in a finite amount of time. Such systems are necessarily discrete. Problems are quite naturally decomposed into sequences of steps - hence our model is sequential. We require that our machine not be subject to uncertainty, hence its behavior is deterministic. There are two finite state machine models : Mealy model - in which outputs occur during transitions.

Moore model - outputs are produced upon arrival at a new state. Mealy Model of FSM Mealy model - transition assigned output  $Q$  = finite set of states  $S$  = input alphabet // the machine's memory // set of stimuli  $R$  =

output alphabet // set of responses = the machine's initial state  $q_1$  : state transition function (or next state function)  $g$  : output function  $g$ : SOR  
 example Design a FSM (Mealy model) which takes in binary inputs and produces a '1' as output whenever the parity of the input string ( so far ) is even.

When designing such models, we should ask ourselves " What is the state set of the machine? ". The state set  $Q$  corresponds to what we need to remember about input strings. We note that the number of possible input strings corresponds to  $I$  which is countably infinite. We observe, however, that a string may have only one of two possible parities. even parity - if  $n_l(w)$  is even. odd parity - if  $n_l(w)$  is odd. And this is all that our machine must remember about a string scanned so far.

Hence  $|Q| = 2$  where  $Q = \{E, o\}$  with  $q_1 = E$  indicating the string has even parity and if  $M_t$  is in state  $o$ , then the string has odd parity. And finally, of course, we must specify the output function  $g$  for this Mealy machine.

According to this machine's specifications, it is supposed to produce an output of '1' whenever the parity of the input string so far is even. Hence, all arcs leading into state  $E$  should be labeled with a '1' output.

Parity Checker (Mealy machine) state diagram Observe our notation that  $g(o, 1) = 1$  is indicated by the arc from state  $o$  to state  $E$  ith a '1' after a slash  
 state table present state input = 0 next state, output input = 1 for this parity machine Observe for the input 10100011 our machine produces the output sequence the corresponding admissible state sequence a second example  
 Construct a Mealy model of an FSM that behaves as a two-unit delay. i. e.  $O$  ,

otherwise A sample input/output session is given below : time 123456789  
 stimulus 001 1 01 00 response 0 0 0 1 1 0 1 Observe that  $r(6) = 1$  which equals  $s(4)$  and so on We know that  $S = R = \{0, 1\}$ . Moore model of FSM Ms  
 " - the output function assigns an output symbol to each state.  $Q =$  finite set of internal states  $S =$  finite input alphabet  $R =$  finite output alphabet  $f :$  state transition function  $h :$  output function  $q_1 = EQ$  is the initial state Design a Moore machine that will analyze input sequences in the binary alphabet  $S = \{0, 1\}$ . Let  $w = s(1) s(2) s(t)$  be an input string  $NO(w) =$  number of O's in  $w$   $NI(w) =$  number of I's in  $w$  then we have that  $IWI = NO(w) + NI(w) =$  The last output of Ms should equal :  $r(t) = [NI(W) \text{ So naturally, the output alphabet } R = \{0, - NO(w)\} \text{ mod } 4.$  stimulus 1 1 01 1 1 00 response 0 1 2 1 23 0 3 2  
 Observe that the length of the output sequence is one longer than the input sequence. Why is this so? Btw : This will always be the case. The corresponding Moore machine : c 2 3 This machine is referred to as an up-down counter.

For the previous input sequence : 11011100 the state sequence is : second example machine should output a '1' whenever this pattern matches the last four inputs, and there has been no overlap, otherwise output a '0'. Hence  $s = R = \{0, 1\}$ . Here is a sample input/output sequence for this machine : 12345678910 11 12 s 101 We observe that 1 because  $s(2) s(3) s(4) s(5)$  however  $r(8) = 0$  because there has been overlap stnce  $s(8) s(9) S(10) 1) = 1011$  What is the state set for this machine??? 0101101 000100000010 1011  
 Ask yourself what is it that Ms must remember in order to function correctly.

Machine Identification Problem The following input-output behavior was exhibited by a transition-assigned machine (Mealy machine) Mt known to

contain three states. Find an appropriate state table for M. Is the table unique? 12345678910 11 12 13 14 input 0000100010 1 0 output 01 01 000010 1 0 0 1 This problem is useful in fault detection and fault location experiments with sequential circuits ( i. e. digital circuits with memory ). One designs a computer circuit. Six months (or six years) later, how does one know that the circuit is working correctly?