THEORY OF COMPUTATION

Tutorial Sheet 4

Submitted By: Submitted To:

Name: Gursimar Kaur Prof. Kuljit Kaur

Roll no: 1820036 (Uni.)

Class: D3 CS E3

Q1: Distinguish between Mealy and Moore Machines.

Sol 1:

|  |  |
| --- | --- |
| MEALY MACHINE | MOORE MACHINE |
| 1. Output depends upon present state and present input. | 1. Output depends upon the present state. |
| 1. Generally, it has fewer state than Moore Machine. | 2. It has more states than Mealy Machine. |
| 1. The value of the output function is a function of the transition and the changes, when the input logic on the present state is done. | 3. The value of the output function is a function of the current state and the changes at the clock edges, whenever state changes occur. |
| 1. It react faster to the inputs because it makes use of same clock cycle. | 4. In this, more logic is required to decode the output resulting in more delays, Generally it makes use of only one clock later. |

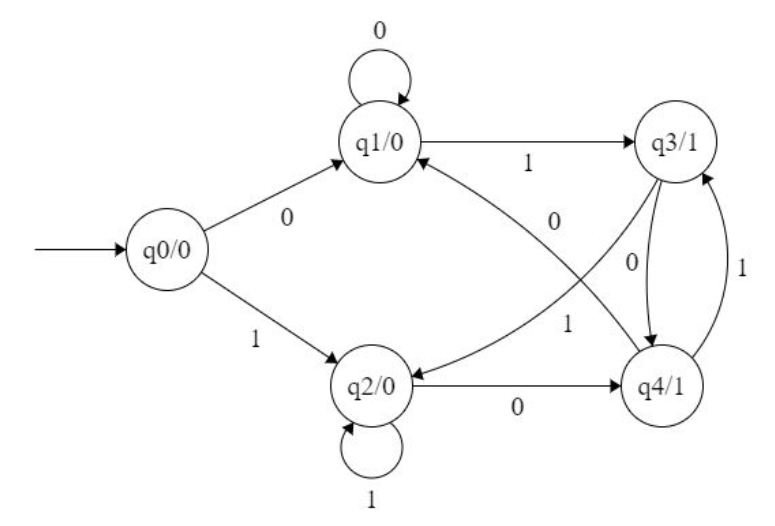


Fig: Example of Moore Machine

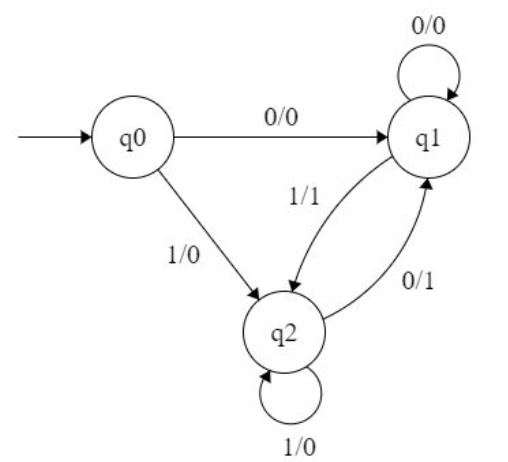


Fig: Example of Mealy Machine

Q2: Prove that each of the classes of languages is closed under union operation.

Sol 2:

Let’s start with the union. For simplicity, let us assume that L1 and L2

are languages over the same alphabet ∑. Since L1 is regular, there exists a DFA M1 = (Q1, ∑, δ, q1, F1) which recognizes L1. Similarly, there exists a DFA M2 = (Q2, ∑, δ, q2, F2) which recognizes L2.

To prove that L1UL2 is regular, we will construct a DFAM∪ which recognizes L1UL2 = {w|w є L1 or w є L2}.

The idea: M∪ = (Q, ∑, δ, q0, F) simulates a parallel execution of M1 and M2. M∪ is

defined as follows:

– Q = Q1 × Q2;

– ∑ is the same;

– δ ((q1, q2), a) = (δ (q1, a), δ (q2, a));

– q0 = (q1, q2);

– F = {(r1, r2) | r1 є F1 or r2 є F2}

To prove correctness we need to show that w є L1 U L2 if and only if M∪ accepts w.

This, in turn, follows from the fact that (r0, r1, . . . , rn) is a computation of M1 on w, and (t0, t1, . . . , tn) is a computation of M2 on w, if and only if ((r0, t0), (r1, t1), . . . , (rn, tn)) is a computation of M∪ on w.

Q3: Show that set of all non-palindromes over {a, b} is a context free language.

Sol 3:

If w is not a palindrome then there must be some i such that the ith letter from the left is different from the ith letter form the right; and vice versa (prove!). This means that

L={Σi−1aΣ∗bΣi−1:i≥1}∪{Σi−1bΣ∗aΣi−1:i≥1},

which is easily seen to be context-free.

Your grammar almost manages to capture this characterization. Unfortunately, it doesn't generate the words ab and ba. A simpler grammar is

S → aSa|bSb|aTb|bTa

T → aT|bT|ϵ

Q4: Identify the languages generated by the following grammars.

a) S -> 0S1 | 0A1 A-> 1A | 1

b) S -> 0S1 | 0A1 A -> 1A0 | 10

Sol 4:

1. i) S -> 0S1

S -> 00S11

S -> 000A111

S -> 0001A111

S -> 00011111

ii) S -> 0S1

S -> 00A11

S -> 001A11

S -> 001111

iii)S -> 0A1

S -> 011

iv)S -> 0A1

S -> 01A1

S -> 011A1

S -> 01111

Language (L) = {w| w is 0n1m such that n>=1 and m>n}

1. S -> 0S1