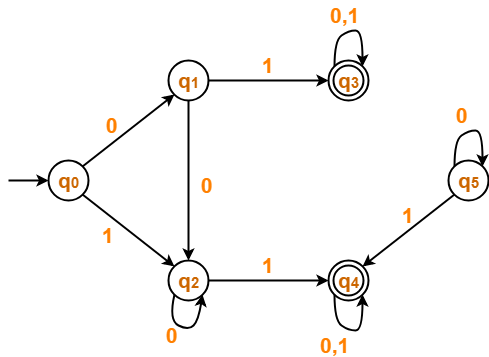
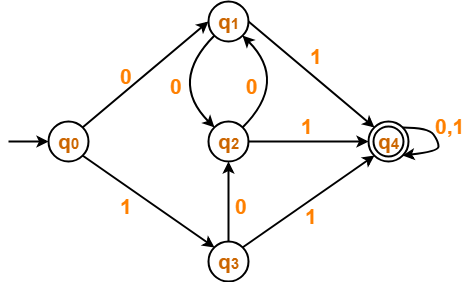
**Minimization of DFA**

* **Minimization of DFA means reducing the number of states from given FA.**
* **Here we reduce the number of states (like unreachable states and dead states).**
* **Also several states are combining based on their next transition states under input column(s).**
* **Purpose is to obtain the FA with less number of states.**
* **Example of unreachable states and dead states (here q5 state is unreachable).**



* **Example of states which has same next transition states under same input column(s).**

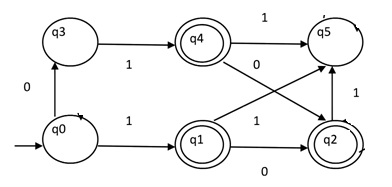


**Check δ (q1, 0) and δ (q3, 0) ????, Check δ (q1, 1) and δ (q3, 1) ????**

* **Example of states which has same next transition states under same input column(s).**

|  |  |  |
| --- | --- | --- |
| State | **Input 0** | **Input 1** |
| →**q0** | q1 | q3 |
| **q1** | q2 | q0 |
| **q2** | q1 | q3 |
| **q3** | q2 | q2 |
|  | q4 | q4 |

* **Two states q1 and q2 are equivalent ( q1 ≡ q2 ) if, both δ (q1, a) and δ (q2, a) are either:- (i) final states or (ii) both of them are non-final states for all a Є Ʃ\*.**
* **Transaction functions {(δ(q, a), δ(q’, a)}, if reach to final states, are 1-equivalent and transaction functions {(δ(q, a), δ(q’, a)}, if reach to non-final states are also 1-equivalent.**



**Check δ (q1, 0) and δ (q4, 0) ???? Check δ (q1, 1) and δ (q4, 1) ????**

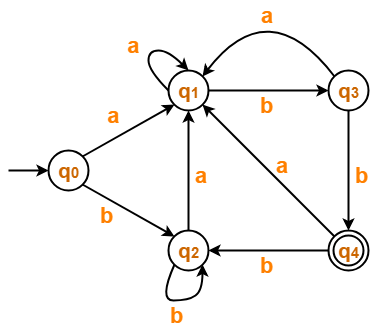
* **We use following notations while processing for minimization of FA.**
* **Q 🡪 set of states.**
* **Π🡪Partition class, Πk 🡪k-equivalent partition class.**

**Steps of construction of minimum automaton are as follows.**

**Step 1: Construction of Π0 :-**

* **We generate 0-equivalent partition class. We will divide Q (set of states) into two sets. One set will contain all final states and other set will contain non final states**
* **This partition is called Π0 = . Where**

**Example 1 :-**



**Transition Table :-**

|  |  |  |
| --- | --- | --- |
| **State** | **Input a** | **Input b** |
| **→q0** | **q1** | **q2** |
| **q1** | **q1** | **q3** |
| **q2** | **q1** | **q2** |
| **q3** | **q1** | **q4** |
|  | **q1** | **q2** |

**Step 1: Construction of Π0 :-**

* **Partition class Π0 or 0-equivalent class.**
* **Since there is only one final state q4, so and**
* **= { q0, q1, q2, q3 }, where Q is set of all states.**
* **Hence partition class Π0 = = { {q4} , { q0, q1, q2, q3 } }**

**Step 2 : Construction of Πk+1 from Πk , where k=0, 1, 2,………**

**Construction of Π1 :-**

* **Partition class Π1 = , where ,since there is only one final state q4.**
* **Contains the set of next transition states (1–equivalent), derived from . Just check their () next state transitions, either next state transitions is (i) final state or (ii) non- final state.**
* **For getting , do the partition of based the next state transitions under the input column(s).**
* **Members of set , for which, next state transitions are non-final states, will be in one set.**
* **Members of set , for which, next state transitions are final states) will be in one set.**
* **Partition of will be as follows,**
* **= { q0, q1, q2 } , { q3 } *(intermediately representation)***
* **Now we need to represent these two set individually. So we take one more set as and finally define = { q0, q1, q2 } and = { q3 }**
* **Accordingly Π1** =  **= { { q4 }, { q0, q1, q2 } , { q3 } }**

|  |  |  |
| --- | --- | --- |
| **State** | **Input a** | **Input b** |
| **→q0** | **q1** | **q2** |
| **q1** | **q1** | **q3** |
| **q2** | **q1** | **q2** |
| **q3** | **q1** | **q4** |
|  | **q1** | **q2** |

**Construction of Π2 :-**

* **Partition class Π2 = , as derived from Π1 and further partition of set { q0, q1, q2 } based on equivalency of next transition states, we take one more set as = { q1 } and finally partition class Π2 is constructed as follows.**
* **Π2** =  **= { { q4 }, { q0, q2 }, { q1 } , { q3 } }**

|  |  |  |
| --- | --- | --- |
| **State** | **Input a** | **Input b** |
| **→q0** | **q1** | **q2** |
| **q1** | **q1** | **q3** |
| **q2** | **q1** | **q2** |
| **q3** | **q1** | **q4** |
|  | **q1** | **q2** |

**Construction of Π3 :-**

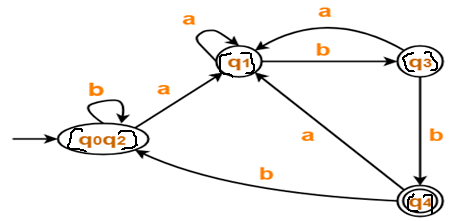
* **Partition class Π3 =**  **as derived from Π2 and here you may notice that no further partition takes place. So Π2 and Π3  are same.**
* **Π2** =  **= { { q4 }, { q0, q2 }, { q1 } , { q3 } }**

|  |  |  |
| --- | --- | --- |
| **State** | **Input a** | **Input b** |
| **→q0** | **q1** | **q2** |
| **q1** | **q1** | **q3** |
| **q2** | **q1** | **q2** |
| **q3** | **q1** | **q4** |
|  | **q1** | **q2** |

* **Step 3: Likewise partition is done, until previous partition class Πk and present partition class Πk+1 become same. Means no further partition is possible.**
* **As** **Π2 and Π3  are same, so we stop here. Finally minimized Finite Automaton M’ = (Q’, Ʃ, δ’, q’0, F’) is as follows, Where Q’= { [q4], [q0, q2], [q1], [q3] } , q’0 = [q0, q2], F’ = [q4] and δ is defined by the table (Transition Table of Minimum State Automaton.**

|  |  |  |
| --- | --- | --- |
| **States / Ʃ** | **Input a** | **Input b** |
| **→ [q0, q2]** | **[q1]** | **[q0, q2]** |
| **[q1]** | **[q1]** | **[q3]** |
| **[q3]** | **[q1]** | **[q4]** |
| **[q4]** | **[q1]** | **[q0, q2]** |

|  |  |  |
| --- | --- | --- |
| **State** | **Input a** | **Input b** |
| **→q0** | **q1** | **q2** |
| **q1** | **q1** | **q3** |
| **q2** | **q1** | **q2** |
| **q3** | **q1** | **q4** |
|  | **q1** | **q2** |



**Example 2 :- For given Finite Automata M = (Q, Ʃ, δ, q0, F), following is the transition diagram. We will first convert into transition table.**

|  |  |  |
| --- | --- | --- |
| **States / Ʃ** | **Input a** | **Input b** |
| **q0** | **q1** | **q0** |
| **q1** | **q0** | **q2** |
| **q2** | **q3** | **q1** |
| **q3** | **q3** | **q0** |
| **q4** | **q3** | **q5** |
| **q5** | **q6** | **q4** |
| **q6** | **q5** | **q6** |
| **q7** | **q6** | **q3** |

**Since there is only one final state q3, so and**

**= { q0, q1, q2, q4, q5, q6, q7 }**

**Hence partition class Π0 = = { {q3} , { q0, q1, q2, q4, q5, q6, q7 } }**

**Step 2 : Construction of Πk+1 from Πk , where k=0, 1, 2,………**

**Construction of Π1 partition class**

**In our example partition class Π1 = is define as follows.**

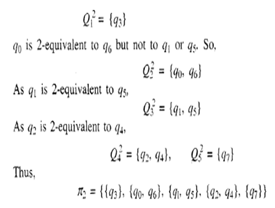
|  |  |  |
| --- | --- | --- |
| **States / Ʃ** | **Input a** | **Input b** |
| **q0** | **q1** | **q0** |
| **q1** | **q0** | **q2** |
| **q2** | **q3** | **q1** |
| **q3** | **q3** | **q0** |
| **q4** | **q3** | **q5** |
| **q5** | **q6** | **q4** |
| **q6** | **q5** | **q6** |
| **q7** | **q6** | **q3** |

* **Partition class Π1 = , where , since there is only one final state q3.**
* **Contains the set of next transition states (1–equivalent), derived from .**
* **To construct , loot at .**
* **Members of set , for which, next state transitions are non-final states, will be in one set. So { q0, q1, q5, q6 } will be in one set.**
* **Members of set , for which, next state transitions are final states, will be in one set. So { q2, q4, q7 } will be in one another set.**
* **So for time being . But it’s not ending here.**
* **Further partitions are continuously done, if found same next state transitions under either all input column(s), or a few input column(s) or belong to same set.**

|  |  |  |
| --- | --- | --- |
| **States / Ʃ** | **Input a** | **Input b** |
| **q0** | **q1** | **q0** |
| **q1** | **q0** | **q2** |
| **q2** | **q3** | **q1** |
| **q3** | **q3** | **q0** |
| **q4** | **q3** | **q5** |
| **q5** | **q6** | **q4** |
| **q6** | **q5** | **q6** |
| **q7** | **q6** | **q3** |

* **Here look at the set = . We found that further partitioned based on their same next state transition under column a.**
* **So set will be finally partitioned as .**
* **Here look at the set = . We don’t found their next state transition under any column is same. No further partition is done here.**
* **So the set .**
* **Partition class Π1 = { q3 }, {q0, q1, q5, q6}, { q2, q4 }, { q7 } }.**

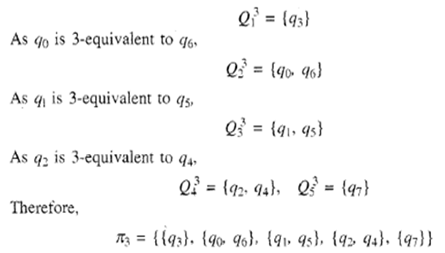
**Next is Construction of Π2 partition class:-**

****

|  |  |  |
| --- | --- | --- |
| **States / Ʃ** | **Input a** | **Input b** |
| **q0** | **q1** | **q0** |
| **q1** | **q0** | **q2** |
| **q2** | **q3** | **q1** |
|  | **q3** | **q0** |
| **q4** | **q3** | **q5** |
| **q5** | **q6** | **q4** |
| **q6** | **q5** | **q6** |
| **q7** | **q6** | **q3** |

**Next is Construction of Π3 partition class:-**

|  |  |  |
| --- | --- | --- |
| **States / Ʃ** | **Input a** | **Input b** |
| **q0** | **q1** | **q0** |
| **q1** | **q0** | **q2** |
| **q2** | **q3** | **q1** |
|  | **q3** | **q0** |
| **q4** | **q3** | **q5** |
| **q5** | **q6** | **q4** |
| **q6** | **q5** | **q6** |
| **q7** | **q6** | **q3** |

****

**Repeat the process of constructing partition Πclasses until Πk+1 = Πk .**

**Like here Π3 = Π2 , so we stop here.**

**And finally minimized Finite Automaton M’ = (Q’, Ʃ, δ’, q’0, F’) is as follows.**

**Where Q’= { [q3], [q0, q6], [q1, q5], [q2, q4], [q7] }**

**q’0 = [q0, q6], F’ = [q3]**

**and δ is defined by the table (Transition Table of Minimum State Automaton).**

**And finally minimized Finite Automaton M’ = (Q’, Ʃ, δ’, q’0, F’) is as follows.**

**Where Q’= { [q3], [q0, q6], [q1, q5], [q2, q4], [q7] }**

**q’0 = [q0, q6], F’ = [q3]**

**and δ is defined by the table (Transition Table of Minimum State Automaton).**

**Transition Table of Minimum State Automaton**

|  |  |  |
| --- | --- | --- |
| **States / Ʃ** | **Input a** | **Input b** |
| **[q0, q6]** | **[q1, q5]** | **[q0, q6]** |
| **[q1, q5]** | **[q0, q6]** | **[q2, q4]** |
| **[q2, q4]** | **[q3]** | **[q1, q5]** |
| **[q3]** | **[q3]** | **[q0, q6]** |
| **[q7]** | **[q0, q6]** | **[q3]** |

**Transition Table of initial Finite Automaton**

|  |  |  |
| --- | --- | --- |
| **States / Ʃ** | **Input a** | **Input b** |
| **q0** | **q1** | **q0** |
| **q1** | **q0** | **q2** |
| **q2** | **q3** | **q1** |
| **q3** | **q3** | **q0** |
| **q4** | **q3** | **q5** |
| **q5** | **q6** | **q4** |
| **q6** | **q5** | **q6** |
| **q7** | **q6** | **q3** |

