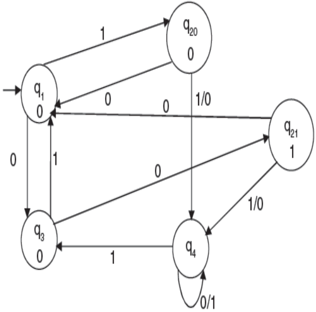

Fig. 3.46

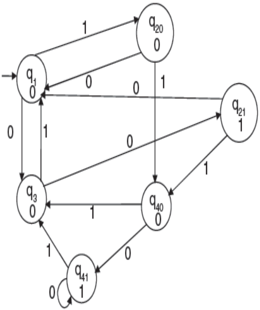

Fig. 3.47

3.14 Minimization of Finite Automata

The language (regular expression) produced by a DFA is always unique. But the reverse, i.e., a language produces a unique DFA, is not true. For this reason, there may be different DFAs in a given language. By minimizing, we can get a minimized DFA with minimum number of states and transitions which produces that particular language. The DFA determines how computers manipulate regular languages (expressions). The DFA size determines the space/time efficiency. So, a DFA with minimized states needs less time to manipulate a regular expression.

3.14 Minimization of Finite Automata

The language (regular expression) produced by a DFA is always unique. But the reverse, i.e., a language produces a unique DFA, is not true. For this reason, there may be different DFAs in a given language. By minimizing, we can get a minimized DFA with minimum number of states and transitions which produces that particular language. The DFA determines how computers manipulate regular languages (expressions). The DFA size determines the space/time efficiency. So, a DFA with minimized states needs less time to manipulate a regular expression.

Before describing the process, we need to know the defi nitions of dead state, inaccessible state, equivalent state, distinguishable state, and k-equivalence in relation with finite automata.

- **Dead State:** A state q_i is called a dead state if q_i is not a final state and for all the inputs to this state, the transitions are confined to that state. In mathematical notation, we can denote $q_i \in F$ and $\delta(q_i, \Sigma) \rightarrow q_i$.
- **Inaccessible State:** The states which can never be reached from the initial state are called inaccessible states.

- **Dead State:** A state q_i is called a dead state if q_i is not a final state and for all the inputs to this state, the transitions are confined to that state. In mathematical notation, we can denote $q_i \in F$ and $\delta(q_i, \Sigma) \rightarrow q_i$.
- **Inaccessible State:** The states which can never be reached from the initial state are called inaccessible states.

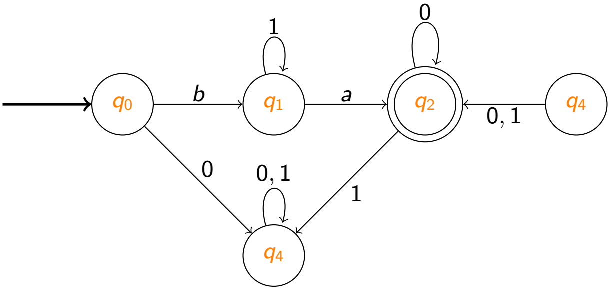

Fig. 3.48

Here, q_d is a dead state and q_a is an inaccessible state as shown in Fig. 3.48.

- **Equivalent state:** Two states q_i and q_j of a finite automata M are called equivalent if both $\delta(q_i, x)$ and $\delta(q_j, x)$ produce final states or both of them produce non-final states for all $x \varepsilon \Sigma *$. It is denoted by $q_i \equiv q_j$.
- **Distinguishable state:** Two states q_i and q_j of a finite automata M are called distinguishable if, for a minimum length string x, for $\delta(q_i, x)$ and $\delta(q_j, x)$, one produces final state and another produces non-final state or vice versa for all $x \varepsilon \Sigma *$.

82 | Introduction to Automata Theory, Formal Languages and Computation

- **K-equivalent:** Two states q_i and q_j of a finite automata M are called k-equivalent ($k >= 0$) if both $\delta(q_i, x)$ and $\delta(q_j, x)$ produce final states or both of them produce non-final states for all $x \in \Sigma*$ of length k or less.

3.14.1 Process of Minimizing

- All the states are '0' equivalent. Mark this as S_0.
- Divide the set of states into two subsets: set of final states and set of non-final states. Mark this as S_1.
- Apply all the inputs separately on the two subsets and find the next state combinations. If it happens that, for applying input on one set of states, the next states belong to different subsets, then separate the states which produce next states belonging to different subsets.
- Continue step (3) for $n+1$ times.
- If S_n and $S_n + 1$ are the same, then stop and declare S_n as an equivalent partition.
- Mark each subset of S_n as a different state and construct the transitional table accordingly. This is the minimum automata.

Consider the following examples to make the minimization process clear.

Example 3.25 Construct a minimum state automaton from the transitional table given below.

Consider the following examples to make the minimization process clear.

Example 3.25 Construct a minimum state automaton from the transitional table given below.

Present State	Next State I/P = 0	Next State I/P = 1
→ q_0	q_1	q_2
q_1	q_2	q_3
q_2	q_2	q_4
q_3	q_3	q_3
q_4	q_4	q_4
q_5	q_5	q_4

Solution: In the finite automata, the states are $\{q_0, q_1, q_2, q_3, q_4, q_5\}$. Name this set as S_0.

$$S_0 : \{q_0, q_1, q_2, q_3, q_4, q_5\}$$

All of the states are 0 equivalents.

In the finite automata, there are two types of states: final state and non-final states. So, divide the set of states into two parts, Q_1 and Q_2.

$$Q_1 = \{q_0, q_1, q_2\} Q_2 = \{q_3, q_4, q_5\}$$

$$S_1 : \{\{q_0, q_1, q_2\}, \{q_3, q_4, q_5\}\}$$

The states belonging to the same subset are 1-equivalent because they are in the same set for string length 1. The states belonging to different subsets are 1-distinguishable.

For input 0 and 1, q_0 goes to q_1 and q_2, respectively. Both of the states belong to the same subset. For q_1 and q_2 with input 0 and 1, the next states are q_2, q_3 and q_2, q_4, respectively. For both of the states, for input 0, the next state belongs to one subset, and for input 1, the next state belongs to another subset. So, q_0 can be distinguished from q_1 and q_2.

The next states with input 0 and 1 for states q_3, q_4, and q_5 belong to the same subset. So, they cannot be divided.

The states belonging to the same subset are 1-equivalent because they are in the same set for string length 1. The states belonging to different subsets are 1-distinguishable.

For input 0 and 1, q_0 goes to q_1 and q_2, respectively. Both of the states belong to the same subset. For q_1 and q_2 with input 0 and 1, the next states are q_2, q_3 and q_2, q_4, respectively. For both of the states, for input 0, the next state belongs to one subset, and for input 1, the next state belongs to another subset. So, q_0 can be distinguished from q_1 and q_2.

The next states with input 0 and 1 for states q_3, q_4, and q_5 belong to the same subset. So, they cannot be divided.

q_0 is the single state in the subset. So, it cannot be divided.

For states q_1 and q_2 with input 0 and 1, for both of the cases, one state belongs to one subset and another state belongs to another subset. So, they cannot be divided.

The next states with input 0 and 1 for states q3, q4, and q5 belong to the same subset. So, they cannot be divided.

So, in the next step,

q_0 is the single state in the subset. So, it cannot be divided.

For states q_1 and q_2 with input 0 and 1, for both of the cases, one state belongs to one subset and another state belongs to another subset. So, they cannot be divided.

The next states with input 0 and 1 for states q3, q4, and q5 belong to the same subset. So, they cannot be divided.

So, in the next step,

$$S_3 : \{\{q_0\}, \{q_1, q_2\}, \{q_3, q_4, q_5\}\}$$

S_2 and S_3 are equivalent.

As step (n–1) and step n are the same, there is no need of further advancement.

In the minimized automata, the number of states is 3.

The minimized finite automata is presented in tabular format as follows:

S_2 and S_3 are equivalent.

As step (n–1) and step n are the same, there is no need of further advancement.

In the minimized automata, the number of states is 3.

The minimized finite automata is presented in tabular format as follows:

	NextState	
State	$I/P = 0$	$I/P = 1$
$\{q_0\}$	$\{q - 1\}$	$\{q_2\}$
$\{q - 1, q_2\}$	$\{q_2\}$	$\{q_3, q_4\}$
$\{q_3, q_4, q_5\}$	$\{q_3, q_4, q_5\}$	$\{q_3, q_4, q_5\}$

But $\{q_1\}$, $\{q_2\}$, and $\{q_3, q_4\}$ do not exist under the column of present state. They are not states of the minimized finite automata, but they are subset of the states. In the next state columns, by replacing the subsets by proper state, the modified table becomes

But $\{q_1\}$, $\{q_2\}$, and $\{q_3, q_4\}$ do not exist under the column of present state. They are not states of the minimized finite automata, but they are subset of the states. In the next state columns, by replacing the subsets by proper state, the modified table becomes

	NextState	
State	$I/P = 0$	$I/P = 1$
$\{q_0\}$	$\{q-1, q_2\}$	$\{q-1, q_2\}$
$\{q-1, q_2\}$	$\{q-1, q_2\}$	$\{q_3, q_4, q_5\}$
$\{q_3, q_4, q_5\}$	$\{q_3, q_4, q_5\}$	$\{q_3, q_4, q_5\}$

But $\{q_1\}$, $\{q_2\}$, and $\{q_3, q_4\}$ do not exist under the column of present state. They are not states of the minimized finite automata, but they are subset of the states. In the next state columns, by replacing the subsets by proper state, the modified table becomes

	NextState	
State	$I/P = 0$	$I/P = 1$
$\{q_0\}$	$\{q-1, q_2\}$	$\{q-1, q_2\}$
$\{q-1, q_2\}$	$\{q-1, q_2\}$	$\{q_3, q_4, q_5\}$
$\{q_3, q_4, q_5\}$	$\{q_3, q_4, q_5\}$	$\{q_3, q_4, q_5\}$

As q_0 is the beginning state of the original finite automata, $\{q_0\}$ will be the beginning state of minimized finite automata. As q_3, q_4, and q_5 are the final states of the original finite automata, the set of the states containing any of the states as element is final state. Here, all the states are contained in a single set $\{q_3, q_4, q_5\}$ and, therefore, it is the final state. By replacing $\{q_0\}$ as A, $\{q_1, q_2\}$ as B, and $\{q_3, q_4, q_5\}$ as C, the modified minimized finite automata becomes

84 | Introduction to Automata Theory, Formal Languages and Computation

State	Next State	
	I/P = 0	I/P = 1
→A	B	B
B	B	C
C	C	C

84 | Introduction to Automata Theory, Formal Languages and Computation

State	Next State	
	I/P = 0	I/P = 1
→A	B	B
B	B	C
C	C	C

The transitional diagram of the minimized finite automata is given in Fig. 3.49.

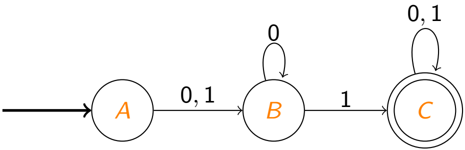

Fig. 3.49

Example 3.26 Construct a minimum state automaton from the following transitional table.

Example 3.26 Construct a minimum state automaton from the following transitional table.

	NextState	
PresentState	$I/P = 0$	$I/P = 1$
A	F	B
B	C	G
C	C	A
D	G	C
E	F	H
F	G	C
G	E	G
H	C	G

A is initial state and C is final state

Solution: In the Finite automata, the states are $\{A, B, C, D, E, F, G, H\}$.

Name this set as S_0.

Solution: In the Finite automata, the states are $\{A, B, C, D, E, F, G, H\}$. Name this set as S_0.

$$S_0 : \{A, B, C, D, E, F, G, H\}$$

All of the states are 0 equivalents.

In the finite automata, there are two types of states: final state and non-final state. The set of states is divided into two parts, namely, Q_1 and Q_2.

Solution: In the Finite automata, the states are $\{A, B, C, D, E, F, G, H\}$. Name this set as S_0.

$$S_0 : \{A, B, C, D, E, F, G, H\}$$

All of the states are 0 equivalents.

In the finite automata, there are two types of states: final state and non-final state. The set of states is divided into two parts, namely, Q_1 and Q_2.

$$Q_1 = \{C\} Q_2 = \{A, B, D, E, F, G, H\}$$
$$S_1 : \{\{C\}\{A, B, D, E, F, G, H\}\}$$

Solution: In the Finite automata, the states are $\{A, B, C, D, E, F, G, H\}$. Name this set as S_0.

$$S_0 : \{A, B, C, D, E, F, G, H\}$$

All of the states are 0 equivalents.

In the finite automata, there are two types of states: final state and non-final state. The set of states is divided into two parts, namely, Q_1 and Q_2.

$$Q_1 = \{C\} Q_2 = \{A, B, D, E, F, G, H\}$$
$$S_1 : \{\{C\}\{A, B, D, E, F, G, H\}\}$$

The states belonging to the same subset are 1-equivalent because they are in the same set for string length 1. The states belonging to different subsets are 1-distinguishable.

C is a single state, and so it cannot be divided. Among the states $\{A, B, D, E, F, G, H\}$ for $\{B, D, F, H\}$, for an input of either 0 or 1, the next state belongs to $\{C\}$ which is a different subset (from B with input 0 goes to C, from D with input 1 goes to C, from F with input 1 goes to C, and from H with input 0 goes to C).