This is the LaTeXassignment written by Ali Akbar Khan Mohammadi,
Introduction to Automata Theory, Formal Languages and Computation,
pages 89-92

	Next State	
Present State	a	b
{A}	{B, E}	{B, E}
{B, E}	{C, D, F, G}	{C, D, F, G}
{C, D, F, G}	{H, I}	{H, I}
{H, I}	{H, I}	{H, I}

q_0' : {A}
F' : {C, D, F, G}

Example 3.28: Minimize the following finite automata by the Myhill±Nerode theorem.

	Next State	
Present State	I/P= a	I/P= b
$\rightarrow$ A	B	F
B	A	F
C	G	A
D	H	B
E	A	G
F	H	C
G	A	D
H	A	C

Here F, G, H are final states.

Solution:

Step I: Divide the states of the DFA into two subsets: fi nal (F) and non-final (Q-F).

F= {E, F, G}, Q-F = {A, B, C, D}

Make a two-dimensional matrix (Fig. 3.53) labelled at the left and bottom by the states of the DFA.

A	—	—	—	—	—	—	—	—
B		—	—	—	—	—	—	—
C			—	—	—	—	—	—
D				—	—	—	—	—
E					—	—	—	—
F						—	—	—
G							—	—
H								—
	A	B	C	D	E	F	G	H

Figure 1: Fig. 3.53

Step II:

1. The following combinations are the combination of the beginning and final states.

   (A, E), (A, F), (A, G), (B, E), (B, F), (B, G), (C, E), (C, F), (C, G), (D, E), (D, F), (D, G)

   Put X in these combinations of states. The modified matrix is given in Fig. 3.54.

B							
C							
D							
E							
F	X	X	X	X	X		
G	X	X	X	X	X		
H	X	X	X	X	X		
	A	B	C	D	E	F	G

Figure 2: Fig. 3.54

2. The pair combination of non-final states are (A, B), (A, C), (A, D), (A, E), (B, C), (B, D), (B, E), (C, D), (C, E), and (D, E).

r= δ(A, a) $\rightarrow$ B s = δ(B, a) $\rightarrow$ A, in the place of (A, B), there is neither X nor x. So, in the place of (A, B), there will be 0.

Similarly,

(r, s) = δ((A, C), a) $\rightarrow$ (B, G) (there is X). In the place of (A, C), there will be x.
(r, s) = δ((A, D), a) $\rightarrow$ (B, H) (there is X). In the place of (A, D), there will be x.
(r, s) = δ((A, E), a) $\rightarrow$ (B, A) (there is neither X nor x). In the place of (A, E), there will be 0.
(r, s) = δ((B, C), a) $\rightarrow$ (A, G) (there is X). In the place of (B, C), there will be x.
(r, s) = δ((B, D), a) $\rightarrow$ (A, H) (there is X). In the place of (B, D), there will be x.
(r, s) = δ((B, E), a) $\rightarrow$ (A, A) (there is neither X nor x, only dash). In the place of (B, E), there will be 0.
(r, s) = δ((C, D), a) $\rightarrow$ (G, H) (there is neither X nor x). In the place of (C, D), there will be 0.
(r, s) = δ((C, E), a) $\rightarrow$ (G, A) (there is X). In the place of (C, E), there will be x.
(r, s) = δ((D, E), a) $\rightarrow$ (H, A) (there is X). In the place of (D, E), there will be x.

3. The pair of combinations of final states are (F, G), (F, H), and (G, H).
   (r, s) = δ((F, G), a) $\rightarrow$ (A, H) (there is X). In the place of (F, G), there will be x.
   (r, s) = δ((F, H), a) $\rightarrow$ (H, A) (there is X). In the place of (F, H), there will be x.
   (r, s) = δ((G, H), a) $\rightarrow$ (A, A) (there is neither X nor x, there is only dash). In the place of (G, H), there will be 0.

   The modified table is given in Fig. 3.55.

B	0						
C	x	x					
D	x	x	0				
E	0	0	x	x			
F	X	X	X	X	X		
G	X	X	X	X	X	x	
H	X	X	X	X	X	x	0
	A	B	C	D	E	F	G

Figure 3: Fig. 3.55

The combination of entries 0 are the states of the modified machine. The states of the minimized machine are (A, B), (A, E), (B, E), (C, D), (G, H), i.e., (A, B, E), (C, D), (G, H), and (F) (As F is a final state of the machine, it is left in the state combinations).

(A, B, E) for input µa¶ gives the output (A, B, A) and for input µb¶ gives the output (F, F, G), where (F, F) belongs to one set and (G) belongs to another set. So, it will be divided into (A, B), (E). The states of the minimized machines are (A, B),(E), (C, D), (G, H), and (F). Let us name them as S_1, S_2, S_3, S_4, and S_5.

For the minimized machine M′,

$$Q = \{S_1, S_2, S_3, S_4, \text{and } S_5\}$$
$$\Sigma = \{a, b\}$$

State Table (transitional function δ)

	Next State	
Present State	a	b
S_1	S_1	S_5
S_2	S_1	S_4
S_3	S_4	S_1
S_4	S_1	S_3
S_5	S_4	S_3

3.16 Two-way Finite Automata

Till now, we have studied automata where the reading head moves only in one direction, from left to right.

Two-way finite automata are machines which can traverse (read) an input string in both directions (left and right).

A two-way DFA (2DFA) consists of five touples M = $\{Q, \Sigma, \delta, q_0, F\}$ where Q, Σ, q_0, and F are defined like one-way FDA,but here the transitional function σ is a map from $Q \times \Sigma$ to $Q \times (L, R)$. L means left and R means right. Block diagram of a two-way finite automata is shown in Fig. 3.56.

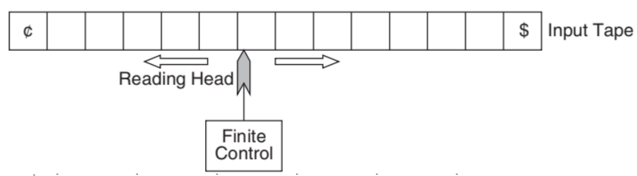

Figure 4: Fig. 3.56: *Two-way finite automata*

Consider the following example of a two-way DFA; M is given in the following table.

	Next State	
Present State	a	b
→ A	A, R	B, R
B	B, R	C, L
C	A, R	C, L

Let us give a string 101001 to check whether it is accepted by the 2DFA or not.

(A, 101001)→ (B, 01001R) → (B, 1001R) → (C, 01001L) → (A, 1001R) → (B, 001R) → (B, 01R)→ (B, 1R) → (C, 01L) → (A, 1R) → B

We have reached the fi nal state, and the string is finished. So, the string 101001 is accepted by the 2DFA.

3.17 Applications of Finite Automata

Finite automata can be applied in different fi elds of computer science and in different engineering fields. Some of them are spelling checkers and advisers, multilanguage dictionaries, minimal perfect hashing, and text compression. Perhaps, the most traditional application is found in compiler construction where such automata can be used to model and implement efficient lexical analysers. A typical example is given in Fig. 3.57 (transitional diagram for relational operators in C).

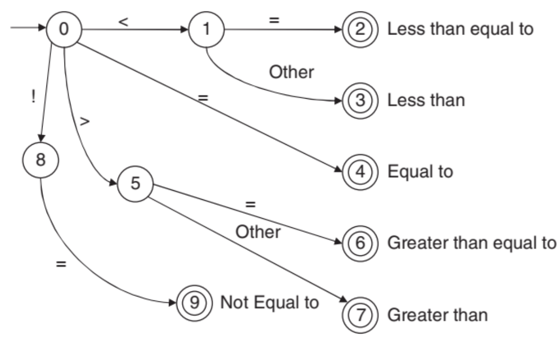

Figure 5: Fig. 3.57: *Transitional Diagram for Relational Operators in C.*

(*Source*: “Compilers: Principles, Techniques and tools” Aho, Sethi, Ullman, Pearson Education.)