

Theory of Automata & Formal Languages (Theory of Computation)

Compiled By
Prof. M. S. Bhatt

Outline

- Regular Expressions
- Regular Languages
- Memory Required for RLs
- State Transition Diagram Construction

More complex examples

(d)

The set of strings over $\{a, b\}$ that contain up to 2 symbols:

(e)

The set of strings over $\{a, b\}$ that begin with a, and have an even number of b:

(f)

The set of strings over $\{a, b\}$ that do not contain the substring aa:

More complex examples

(d)

The set of strings over $\{a, b\}$ that contain up to 2 symbols:

$\lambda + a + b + aa + ab + ba + bb$

(e)

The set of strings over $\{a, b\}$ that begin with a, and have an even number of b:

$a(a^*ba^*b)^*a^*$.

(f)

The set of strings over $\{a, b\}$ that do not contain the substring aa:

$b^*(abb^*)^*(\lambda + a)$

More complex examples

- 1) The language of all strings containing exactly two 0's
- 2) The language of all strings containing at least two 0's
- 3) The language of all strings that begin or end with 00 or 11

More complex examples

- 1) The language of all strings containing exactly two 0's
 $1^*01^*01^*$
- 2) The language of all strings containing at least two 0's
 $(0+1)^*0(0+1)^*0(0+1)^*$
- 3) The language of all strings that begin or end with 00 or 11
 $(00+11)(0+1)^* + (0+1)^*(00+11)$

More complex examples

- 1) The language of all strings containing both 11 and 010 as substrings
- 2) The language of all strings containing 101 and 010 as substrings

More complex examples

- 1) The language of all strings containing both 11 and 010 as substrings

$$(0+1)^*(11(0+1)^*010 + 010(0+1)^*11)(0+1)^*$$

- 2) The language of all strings containing 101 and 010 as substrings

$$(0+1)^*(101(0+1)^*010 + 010(0+1)^*101 + 1010+0101)(0+1)^*$$

Rewrite rules for regular expressions

Let r , s and t be regular expressions over $\{0,1\}$.

$$\emptyset^* = \lambda$$

$$\lambda^* = \lambda$$

$$(r + \lambda)^+ = r^*$$

$$r^* = r^*(r + \lambda) = r^* r^* = (r^*)^*$$

$$(r^* s^*)^* = (r + s)^*$$

Memory required to recognize RLs

- There must be single pass through input from left to right.
- We make a tentative decision after each input symbol. The processing produces a sequence of decisions, one for each prefix, and the final answer for the string is simply the last of these.

Example: String ending with '0' (One Accepting State)

- Accepted

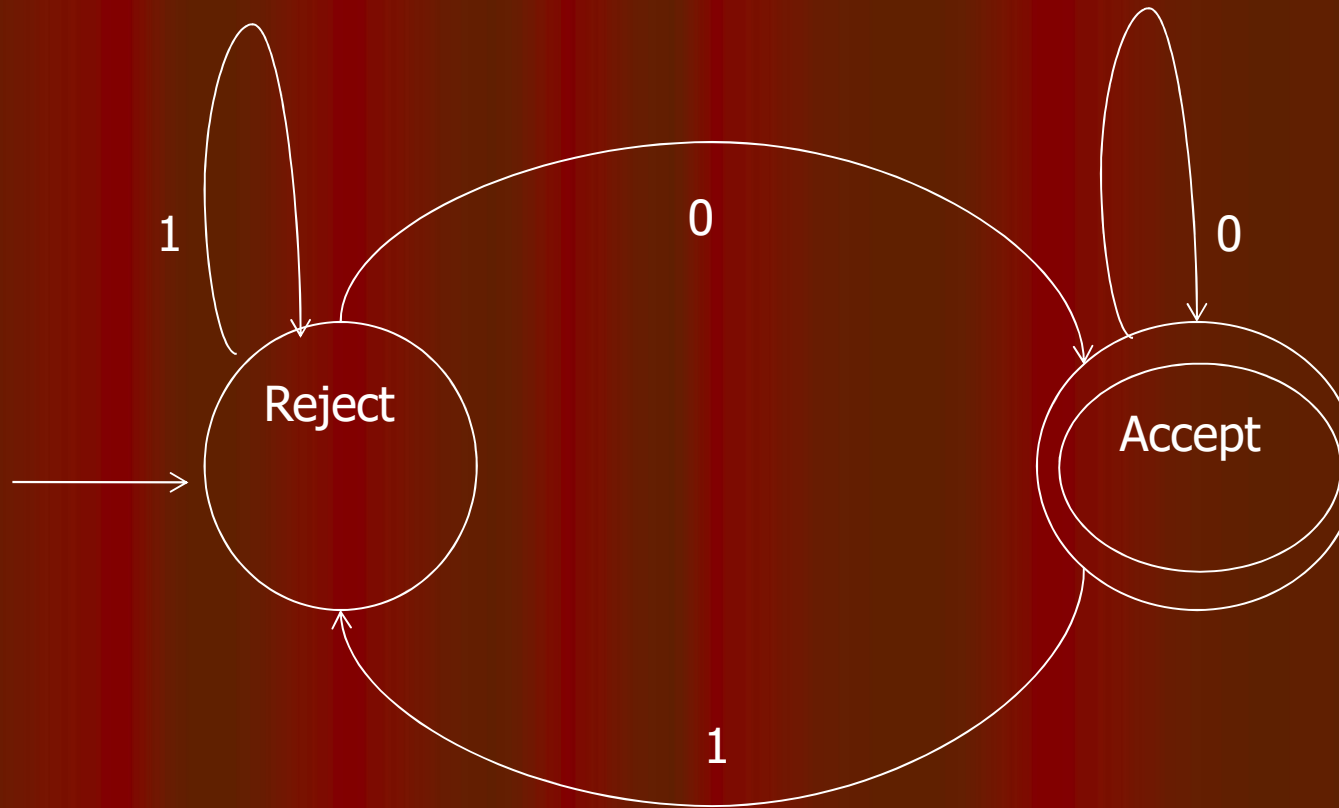
0
00
10
000
010
100
110

- Rejected

Λ
1
01
11
001
011
101
111

How many input symbols are required to change the state from "Accept" to "Reject" / "Reject" to "accept" ?

Example: String ending with '0' (One Accepting State)



Example: String ending with '11' (One Accepting State)

- Accepted

11
011
111

- Rejected

Λ
0
1
00
01
10
000
001
010
100
101
110

How many input symbols are required to change the state from "Accept" to "Reject" / "Reject" to "accept" ?

Example: String ending with '11' (One Accepting State)

- Accepted

11
011
111

- Rejected

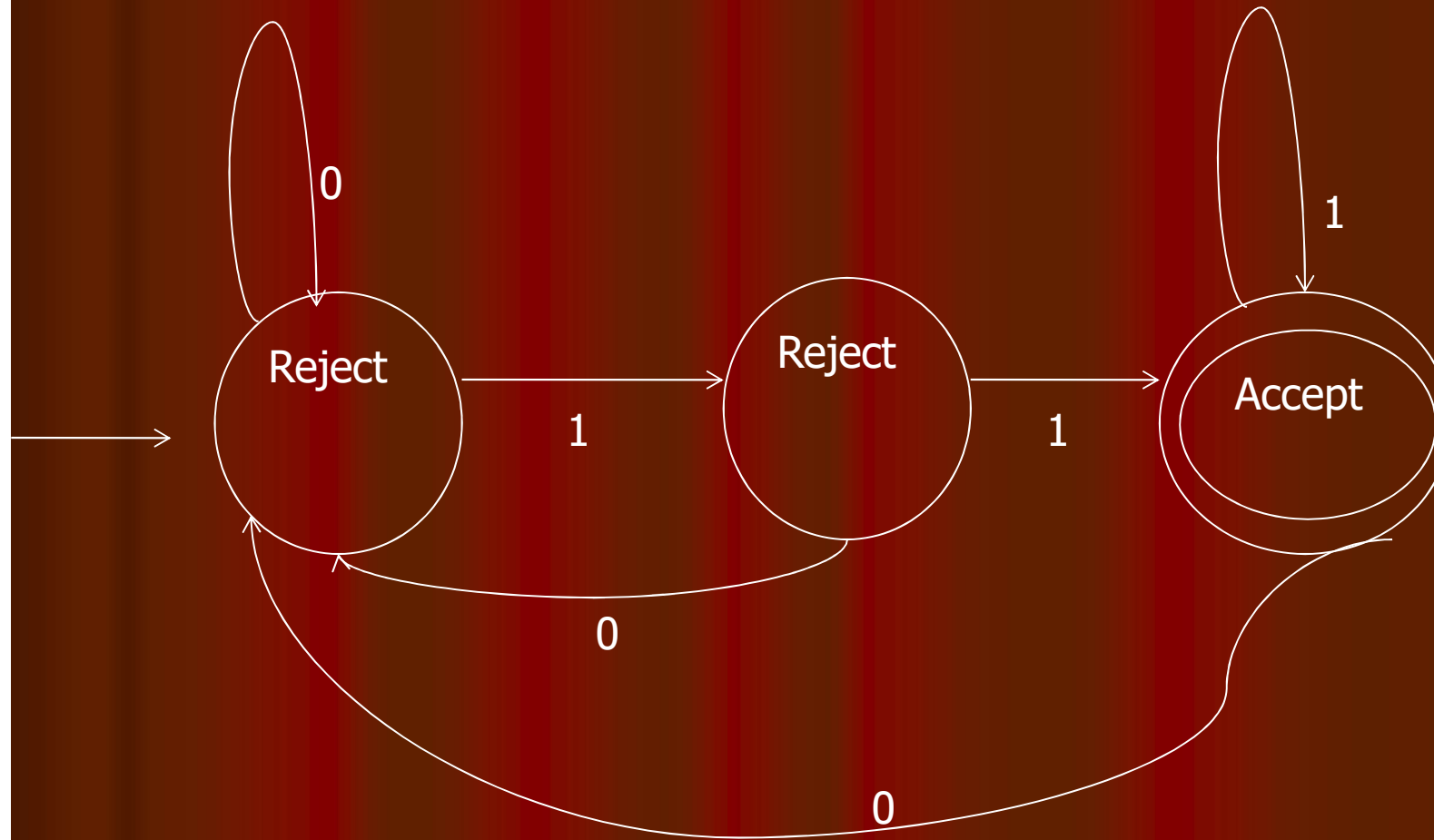
1
01
001
101

- Rejected

Λ
0
00
10
000
010
100
110

How many input symbols are required to change the state from "Accept" to "Reject" / "Reject" to "accept" ?

Example: String ending with '11' (One Accepting State)



Example: String ending with '1' and doesn't contain 00 (One Accepting State)

- Accepted

1
01
11
011
101
111

- Rejected

Λ
0
00
10
000
001
010
100
110

How many input symbols are required to change the state from "Accept" to "Reject" / "Reject" to "accept" ?

Example: String ending with '1' and doesn't contain 00 (One Accepting State)

- Accept

1
01
11
011
101
111

- Reject

0
10
010
110

- Reject

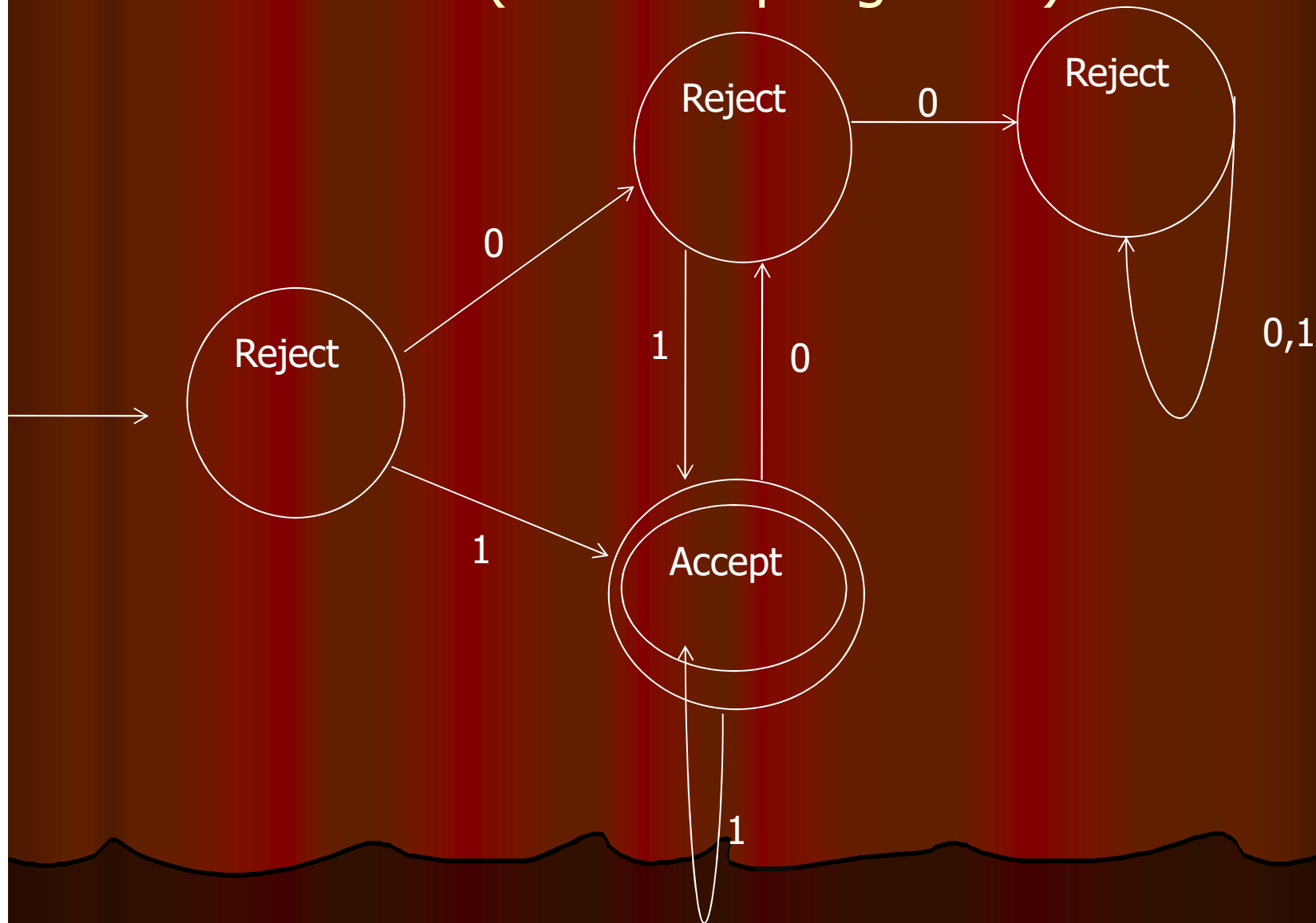
^

- Never Accepted

00
000
001
100

How many input symbols are required to change the state from "Accept" to "Reject" / "Reject" to "accept" ?

Example: String ending with '1' and doesn't contain 00
(One Accepting State)



Example: $\{\alpha \in \Sigma^* \mid \alpha \text{ is a binary number divisible by } 4\}$
(One Accepting State)

$(0+1)^*00$

- Accepted

00
000
100
0000
0100
1000
1100

- Rejected

0
10
010
110

- Rejected

^
1
01
001
101

How many input symbols are required to change the state from "Accept" to "Reject" / "Reject" to "accept" ?

Example: $\{\alpha \in \Sigma^* \mid \alpha \text{ is a binary number divisible by } 4\}$
(One Accepting State)

$(0+1)^*00$

