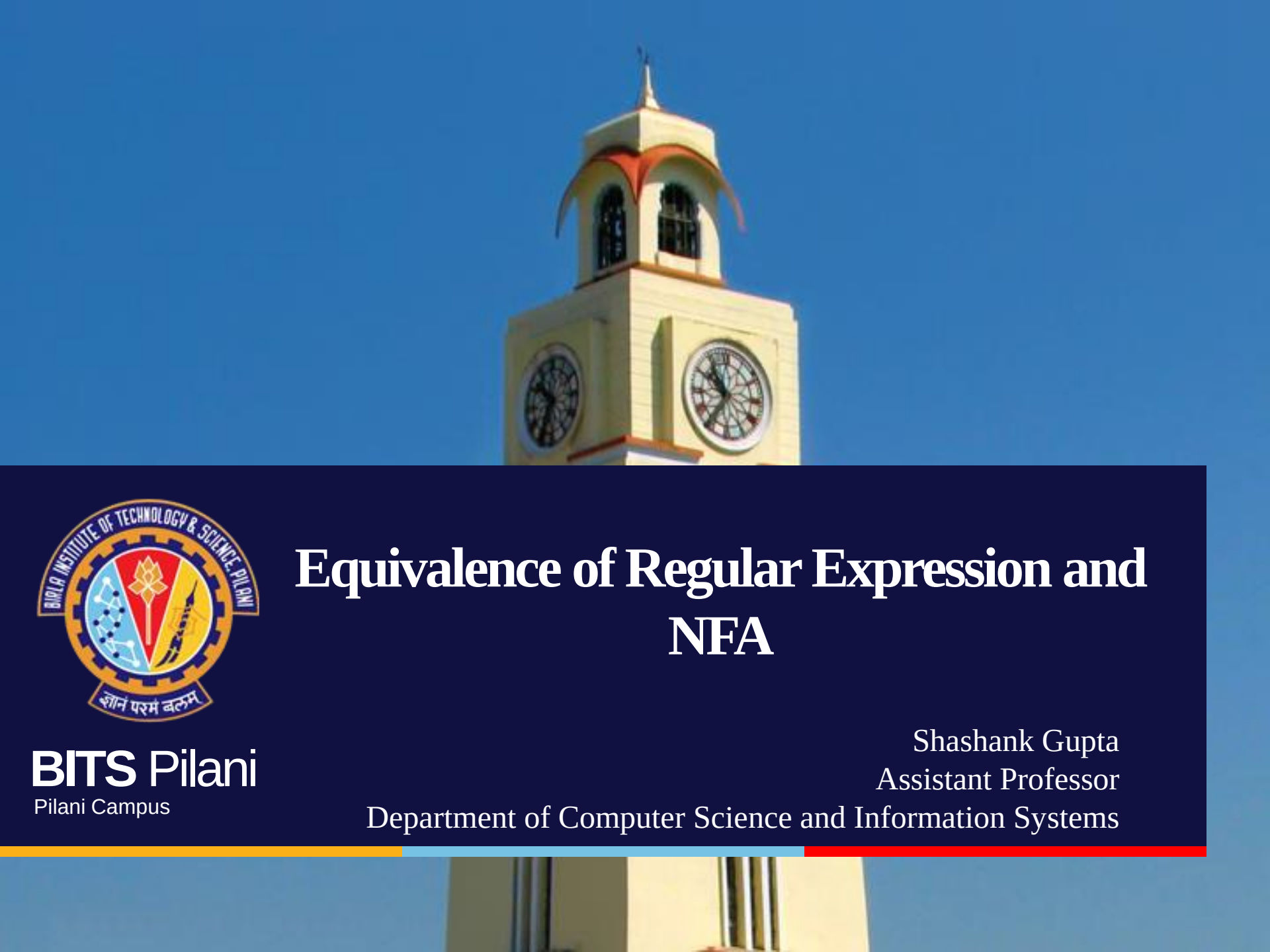


Equivalence of Regular Expression and NFA

BITS Pilani
Pilani Campus

Shashank Gupta
Assistant Professor
Department of Computer Science and Information Systems

Designing of Regular Expression (Cont.…..)

Construct a regular expression that accepts all strings of 0's and 1's where each string has atleast two 0's

Step 1

- L= { 00, 01011, 11100,--}

Step 2

- Minimal length String is 00.

Step 3

- (0+1)*.0.(0+1)*.0.(0+1)*

Designing of Regular Expression (Cont.…..)

Construct a regular expression that accepts all strings of 0's and 1's where each string has '1' at every odd position and the length of the string is odd.

Step 1

- L= { 1, 101, 111, 10111, 10101,-----}

Step 2

- Minimal length String is 1

Step 3

- 1.((0+1).1)*

Algebraic Properties of Regular Expression

- R1 + (R2 + R3) = (R1 + R2) + R3
- R1(R2R3) = (R1R2)R3
- R1(R2 + R3) = R1R2 + R1R3
- (R1 + R2)R3 = R1R3 + R2R3
- R1 + R2 = R2 + R1
- (R1*)* = R1*
- R1.∈ = ∈.R1 = R1
- R1 + Ø = R1

Regular Expressions (REs) and Regular Languages

A language L is regular if and only if L = L(R) for some regular expression R.

REs are equivalent in power to NFAs/DFAs.

Converting RE to NFA

Given a regular expression, we will convert it into a NFA N such that L(R) = L(N).

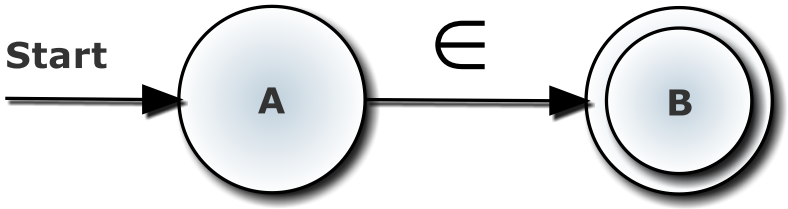
We will give a case based analysis based on the inductive definition of REs.

Case 1

- Case 1: R = ϕ

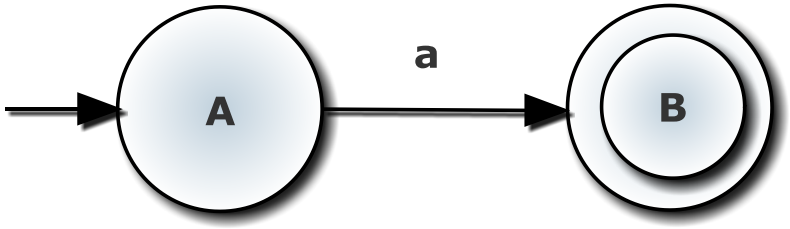
Case 2

- Case 2: R = $\in$



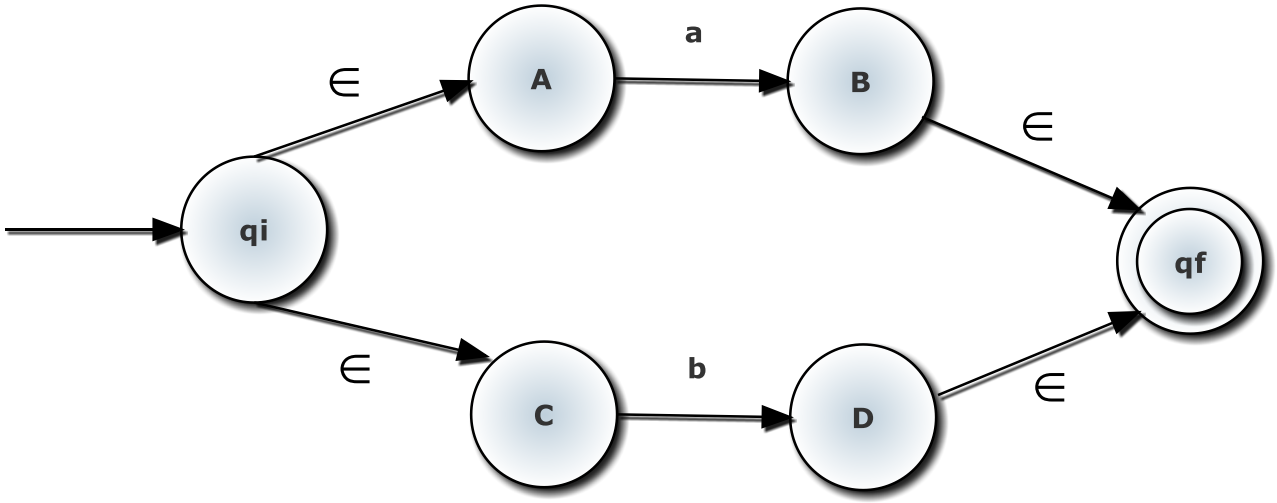
Case 3

- Case 3: R = a



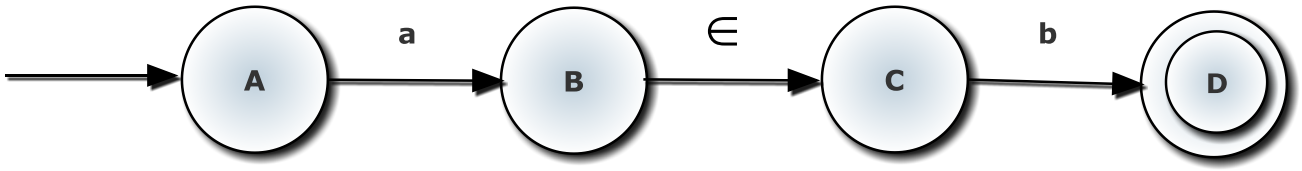
Case 4

- Case 4: R = a + b



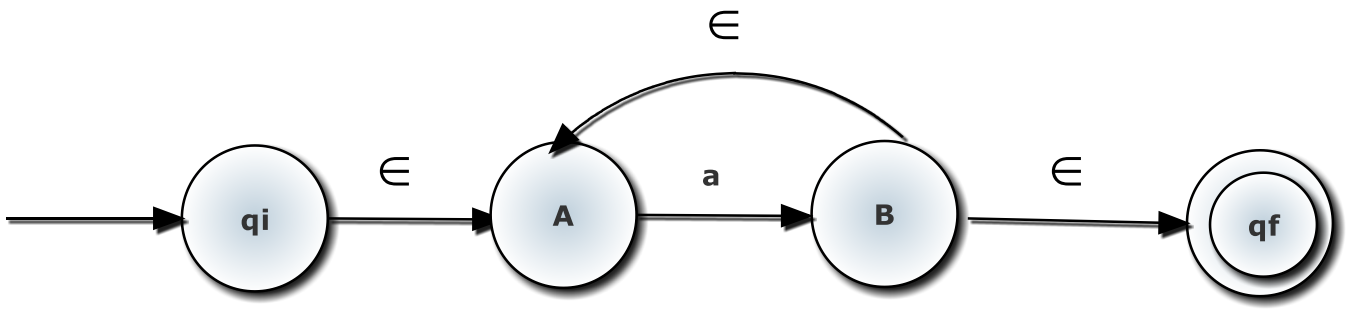
Case 5

- Case 5: R = a.b



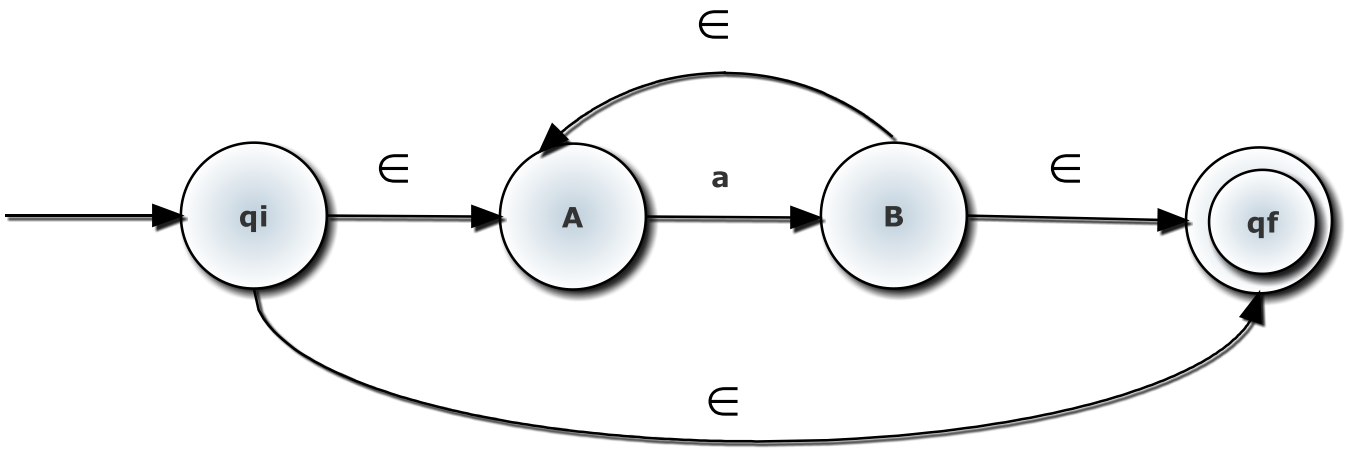
Case 6

- Case 6: $R = a^+$



Case 7

- Case 7: R = a*



Home Assignment

Convert the following regular expression to epsilon NFA

$$(00 + 1)^* 1 (0 + 1)$$