Unit - I | Lecture- 10

Conversion of Regular Expression to NFA

Dr. R. Madana Mohana
Professor, Artificial Intelligence & Data Science | I/c-Head, Artificial Intelligence & Machine Learning
CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
Hyderabad - 500 075, Telangana, INDIA
www.cbit.ac.in

Regular Expressions

- **Finite Automata and Regular Expressions (Constructing Finite Automata for given Regular Expression)**

Finite Automata and Regular Expressions

Equivalence of Finite Automata (FA) and Regular Expressions:

- Given a **regular expression**, there is an associated **Regular Language**. The equivalence of **Finite Automata (FA)** and **Regular Expressions** can be proved by using **Kleen's Theorem**. This theorem has **two** parts:

Finite Automata and Regular Expressions

Equivalence of Finite Automata (FA) and Regular Expressions:

Part-I: Converting Regular Expressions to Automata

"For every regular expression there exists a machine M (Finite Automata) which accepts the regular language. i.e., Regular Expression to Finite Automata (FA)".

Finite Automata and Regular Expressions

Equivalence of Finite Automata (FA) and Regular Expressions:

Part-II: From FA's to Regular Expressions

"The language accepted by Finite Automata (FA) is Regular Language" i.e., Finite Automata (FA) to Regular Expression.

Constructing Finite Automata for given Regular Expression

Part-I: Converting Regular Expressions to Automata

“For every regular expression there exists a machine M (Finite Automata) which accepts the regular language. i.e., Regular Expression to Finite Automata (FA)”.

(or)

“Every language defined by a regular expression is also defined by a finite automata”.

Constructing Finite Automata for given Regular Expression

Procedure:

*For a **regular expression** there exists a **NFA-ϵ** transitions that accepts* $L(R)$.

Step-1: The basis of the construction of an automaton from a regular expression (Zero operators)

The regular expressions for an empty set Φ, an empty string ϵ and an input symbol a, $a \epsilon \Sigma$ are given below:

i) $R = \Phi$

ii) $R = \epsilon$

iii) $R = a,\ a \epsilon \Sigma$

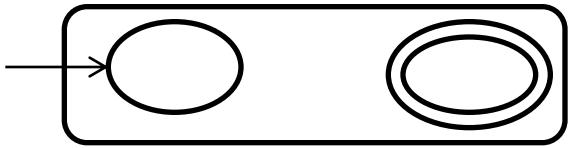
Constructing Finite Automata for given Regular Expression

Step-1: The basis of the construction of an automaton from a regular expression

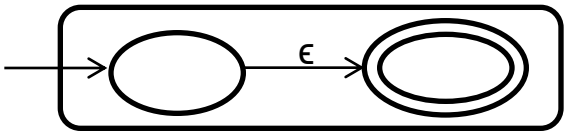
(Zero operators)

NFA's for the following **regular expressions** are given below:

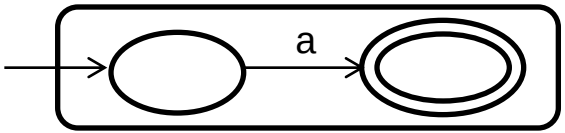
i) **R = Φ**



ii) **R = ϵ**



iii) **R = a, aϵΣ**



Constructing Finite Automata for given Regular Expression

Step-2: The inductive step in the regular expression to NFA-ϵ construction

There are three parts of the induction are shown below:

i)The expression is **R + S (Union)** for some smaller expressions **R** and **S**.

ii)The expression is **RS (Concatenation)** for some smaller expressions **R** and **S**.

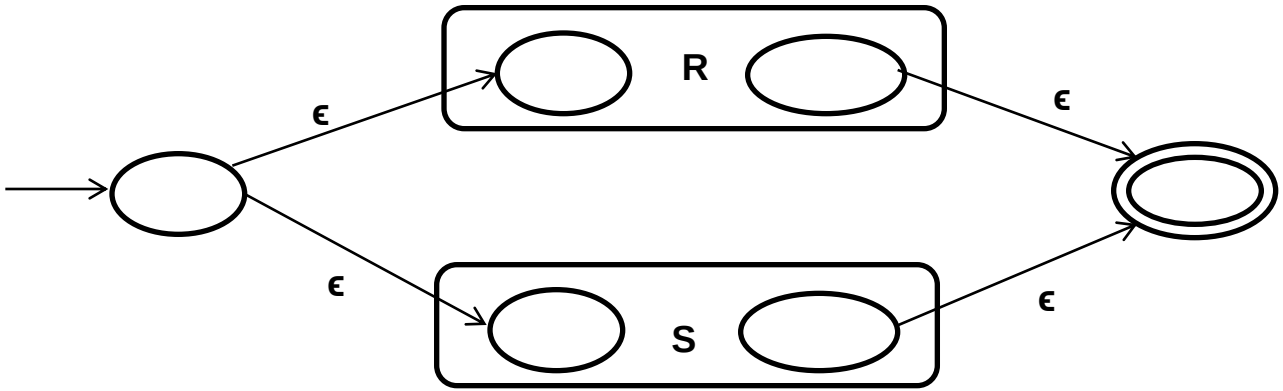
iii)The expression is $\mathbf{R^*}$ **(Closure)** for some smaller expression **R**.

Constructing Finite Automata for given Regular Expression

Step-2: The inductive step in the regular expression to NFA-ϵ construction

NFA's for the following **regular expressions** are given below :

i)The expression is `R + S (Union)` for some smaller expressions `R` and `S`.

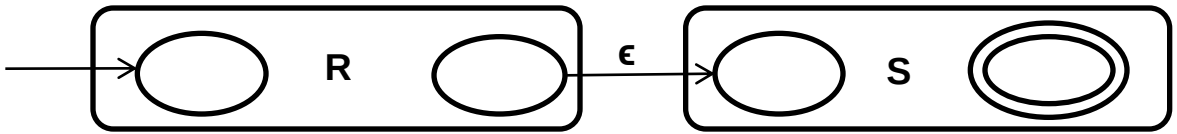


Constructing Finite Automata for given Regular Expression

Step-2: The inductive step in the regular expression to NFA-ϵ construction

NFA's for the following **regular expressions** are given below :

ii)The expression is RS (Concatenation) for some smaller expressions R and S.

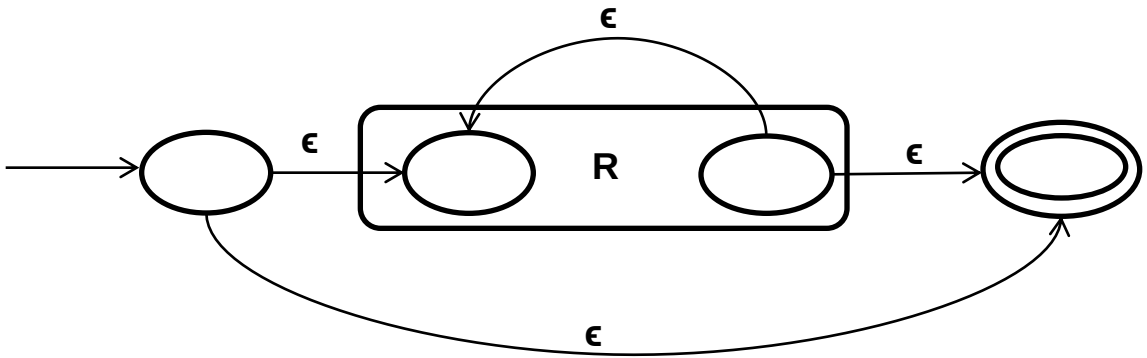


Constructing Finite Automata for given Regular Expression

Step-2: The inductive step in the regular expression to NFA-ϵ construction

NFA's for the following **regular expressions** are given below :

iii)The expression is R^* (Closure) for some smaller expression R.



Constructing Finite Automata for given Regular Expression

Practice Problems:

Q1) Convert the following regular expressions to NFA's with ϵ-transitions:

i. (0+1)*1(0+1)

ii. 01+101

iii. b) $(01+1)^*$

iv. c) $011(0+1)^*$

v. d) $a^*(a+b)ab$

vi. e) 1^*0+0

vii. f) $(0+1)^*(00+11)(0+1)^*$

viii. g) $(00+1)^*(10)^*$

ix. h) 0^*+11

x. i) $(10^*)^*$

xi. j) $((01+001)^*0^*)^*$

Constructing Finite Automata for given Regular Expression

Practice Problems:

Q2) Construct a DFA for the regular expression $(0+1)^*(00+11)(0+1)^*$

Q3) Design a DFA for the following over {a, b}

a) All strings containing not more than three **a**'s

b) All strings that has at least two occurrences of **b** between any two occurrences of **a**.

Q4) Construct a DFA accepting the set of all strings ending with 00?

Q5) Construct a DFA for the Regular Language consisting of any number of a's and b's

- Constructing Finite Automata for given Regular Expression