

Regular expressions and languages

Chapter - 2: Regular languages and finite automata

Prof. Riddhi Atulkumar Mehta

Assistant Professor

Department of Computer Science and
Engineering

Content

1. Regular Expression.....	1
2. Symbols in Regular Expressions.....	2
3. Regular Expression Operators.....	3
4. Examples of Regular Expressions.....	4
5. Language of a Regular Expression.....	5
6. Regular Expressions and Finite Automata.....	6
7. Closure Properties of Regular Languages.....	7
8. Applications of Regular Expressions.....	8

What is a Regular Expression?

Definition:

A Regular Expression (RE) is a formal way to describe a set of strings (a language) using pattern-matching rules.

Purpose:

To specify regular languages, which are the simplest class in the Chomsky hierarchy.

Basic Symbols in Regular Expressions

Symbol	Meaning
a	The character 'a'
ϵ	The empty string
\emptyset	Empty language (no string)

Regular Expression Operators

Operator	Syntax	Description
Union	$R_1 + R_2$	Either R_1 or R_2 (alternation)
Concatenation	$R_1 R_2$	R_1 followed by R_2
Kleene Star	R^*	Zero or more repetitions of R
Plus	R^+	One or more repetitions of R
Optional	$R?$	Zero or one occurrence of R

Examples of Regular Expressions

- $a^* \rightarrow \{\epsilon, a, aa, aaa, \dots\}$
- $(a + b)^* \rightarrow$ All strings over $\{a, b\}$
- $a(b + c)^* \rightarrow$ Starts with 'a', followed by any combo of b and c
- $1(0 + 1)^* \rightarrow$ Binary strings starting with 1

Language of a Regular Expression

Definition:

If R is a regular expression, then $L(R)$ is the language denoted by R — the set of all strings it matches.

Example:

$$R = (ab)^*$$

$$\rightarrow L(R) = \{\epsilon, ab, abab, ababab, \dots\}$$

Regular Expressions and Finite Automata

Every regular expression corresponds to a Finite Automaton, and vice versa.

- $RE \rightarrow FA$ (Construction possible)
- $FA \rightarrow RE$ (Using state elimination method)

Implication:

REs and FAs recognize the same class of languages: Regular Languages

Closure Properties of Regular Languages

Operation	Explanation
Union	$L_1 \cup L_2$ is regular
Concatenation	$L_1 L_2$ is regular
Kleene Star	L^* is regular
Complementation	If L is regular, so is its complement
Intersection	$L_1 \cap L_2$ is regular
Difference	$L_1 - L_2$ is regular

Applications of Regular Expressions

- Text editors (search and replace)
- Lexical analyzers (tokenization in compilers)
- Pattern matching in programming (e.g., Python, JavaScript, grep)
- Validation (email, password formats)

Parul[®]
University

NAAC
GRADE **A++**



<https://paruluniversity.ac.in/>

