

Regular Expression

Compact Study Guide

Course Code: CSE331

Course Title: Automata and Computability

Course Instructor: Tanhiat Fatema Afnan [TNF]

Prepared By: Md. Foysal Ferdous

Department of Computer Science and Engineering
BRAC University

Contents

1 Introduction 2

2 Core Components of Regex 2

2.1 Basic Operators 2

2.2 Special Patterns 2

3 Extended Concepts 2

3.1 Regex Algebra Laws 2

3.2 Identity Patterns 2

4 Closure Properties of Regular Languages 3

5 Finite Automata Regex Summary 3

5.1 Regex → NFA (Thompson Construction) 3

5.2 NFA → DFA 3

5.3 DFA → Regex 3

6 Regex Simplification Techniques 3

7 Common Exam Templates 3

8 Pitfalls Students Often Face 4

9 25 High-Value Exam-Style Regex Problems 4

10 Final Tips 4

1 Introduction

Regular Expressions (regex) describe patterns over strings and define Regular Languages. They are equivalent in power to DFAs, NFAs, and ϵ -NFAs. This guide provides a concise but dense overview for mastering CSE331 Automata questions.

We use flat notation: R_1R_2 (concatenation), $R_1 \mid R_2$ (union), R^* (Kleene star), $*$ (all strings).

2 Core Components of Regex

2.1 Basic Operators

- **Union:** $R_1 \mid R_2$
- **Concatenation:** R_1R_2
- **Kleene Star:** R^*
- **Kleene Plus:** R^+
- **Empty String:** ϵ
- **Empty Language:** \emptyset

2.2 Special Patterns

- Contains substring: $*x^*$
- Avoid substring ab : b^*a^*
- Exactly k occurrences of a : $b^* a b^* a b^* \dots (k \text{ times}) \dots b^*$
- No consecutive aa : $(b \mid ab)^*(\mid a)$
- Even $\#a$: $b^*(a b^* a b^*)^* b^*$
- Alternating: $(ab \mid ba)^*(a \mid b \mid)$

3 Extended Concepts

3.1 Regex Algebra Laws

Associativity:

$$(R_1R_2)R_3 = R_1(R_2R_3)$$

Commutativity of Union:

$$R_1 \mid R_2 = R_2 \mid R_1$$

Distributive Laws:

$$R_1(R_2 \mid R_3) = R_1R_2 \mid R_1R_3$$

$$(R_1 \mid R_2)R_3 = R_1R_3 \mid R_2R_3$$

Idempotent Laws:

$$R \mid R = R$$

Identity:

$$R \mid \emptyset = R, \quad R\epsilon = R$$

3.2 Identity Patterns

- $* x *$ is the universal “contains x ”.
- $(x)^* = \epsilon \mid x \mid xx \mid xxx \mid \dots$

4 Closure Properties of Regular Languages

Regular languages (and regex) are closed under:

- Union
- Intersection
- Complement
- Difference
- Concatenation
- Kleene star and plus
- Reverse
- Homomorphism / Inverse Homomorphism

These let you combine regex constraints logically.

5 Finite Automata Regex Summary

5.1 Regex \rightarrow NFA (Thompson Construction)

General rules:

- Single character: 2 states with transition.
- Concatenation: join NFAs.
- Union: new start/end with ε branches.
- Kleene Star: loopback with ε -transitions.

5.2 NFA \rightarrow DFA

Subset construction (powerset method).

5.3 DFA \rightarrow Regex

State elimination method:

1. Add new start and end.
2. Remove intermediate states.
3. Update edges using:

$$R_{new} = R_{old} | R_{ik}(R_{kk})^* R_{kj}$$

6 Regex Simplification Techniques

- Remove redundant unions: $a \mid a = a$
- Remove dead expressions: $a =$
- Collapse star of star: $(R^*)^* = R^*$
- Combine concatenations: $a = a$

7 Common Exam Templates

- Ends with 01: $*01$
- Starts with 1: $1*$
- Contains exactly one ab: $(\text{no-ab}) \text{ ab } (\text{no-ab})$
- At least two 1s: $*1*1*$
- Odd length: $()^*$

8 Pitfalls Students Often Face

- Confusing a^*b^* with $(ab)^*$
- Forgetting that regex cannot count infinitely (except parity, mod patterns)
- Thinking regex can compare symbols far apart—it cannot
- Using $*$ incorrectly (it means ANY string)

9 25 High-Value Exam-Style Regex Problems

1. Contains aa: $*aa*$
2. Avoid bb: $(a|ba)^*(|b)$
3. Start & end same: $a^*a|b^*b$
4. Exactly two a's: $b^*ab^*ab^*$
5. At least one b: $*b^*$
6. No a's: b^*
7. Even length: $(aa|ab|ba|bb)^*$
8. Every a followed by b: $(b|ab)^*$
9. Ends with aba: $*aba$
10. Avoid ab: b^*a^*
11. Exactly one aa: $(b|ab)^*aa(b|ba)^*$
12. a divisible by 3: $b^*(ab^*ab^*ab^*)^*b^*$
13. Last two different: $*(ab|ba)$
14. Start b, odd a's: $b(b^*ab^*(ab^*ab^*)^*)$
15. Alternating: $(ab|ba)^*(a|b|)$
16. At least two b's: $*b^*b^*$
17. First three abb: abb^*
18. No consecutive same: $(ab|ba)^*(a|b|)$
19. a only in pairs: $(aa|b)^*$
20. Contains ab and ba: $*ab^*ba^*|*ba^*ab^*$
21. Every 0 \rightarrow 11: $1^*(011)^*1^*$
22. Ends a but avoid aa: $(ba)^*a$
23. Third from last b: $*b$
24. XOR (first XOR last is a): $a^*b|b^*a$

10 Final Tips

- Translate English \rightarrow structure \rightarrow regex.
- Always identify forbidden and required substrings.
- Build mini-DFA mentally when confused.
- Use prefix–body–suffix for exact occurrences.