THEORY OF AUTOMATA & FORMAL LANGUAGES

Introduction to Regular Expressions

Describe entire languages using compact patterns

Symbols → Operators → Patterns → Membership

What is a regular expression?

Fix the alphabet $\Sigma = \{a, b\}$. An expression describes a language.

Elementary expressions

$\emptyset$ describes the empty language.

ε describes the language $\{\varepsilon\}$.

a describes $\{a\}$; b describes $\{b\}$.

An expression is not itself a set.

L(r) denotes the language of r.

Read the expression

r = a

L(r) = {a}

a belongs to L(r).

b, aa and ε do not belong.

Only the single symbol a matches.

A regular expression is a compact description of a set of strings.

Union: choosing alternatives

Use | for union (some textbooks use +).

Definition

r | s means r OR s.

L(r | s) = L(r) ∪ L(s)

a | b describes {a, b}.

ε | a describes {ε, a}.

Choose one complete alternative.

Worked example

r = ab | ba

Choose ab → string ab.

Choose ba → string ba.

L(r) = {ab, ba}

aa and aba are not members.

Union offers alternatives; it does not join them.

Concatenation: joining patterns

Writing expressions side by side joins their strings in order.

Definition

rs means r followed by s.

L(rs) = L(r)L(s)

ab describes {ab}.

ba describes {ba}.

Generally rs and sr differ.

Worked example

r = (a | b)b

Choose a, then append b → ab.

Choose b, then append b → bb.

L(r) = {ab, bb}

a and ba are not members.

Parentheses group alternatives before concatenation.

Kleene star and positive closure

Apply the repetition operator to the expression immediately before it.

Zero or more repetitions

$L(r^*) = L(r)^*$

a^* describes $\{\varepsilon, a, aa, aaa, \ldots\}$.

$(ab)^*$ describes $\{\varepsilon, ab, abab, \ldots\}$.

$(a \mid b)^*$ describes all binary-letter strings.

Zero repetitions produce ε.

One or more repetitions

r^+ is shorthand for rr^*.

a^+ describes $\{a, aa, aaa, \ldots\}$.

$(ab)^+$ describes $\{ab, abab, \ldots\}$.

If $\varepsilon \in L(r)$, then $\varepsilon \in L(r^+)$.

Example: $(\varepsilon \mid a)^+$ includes ε.

Star repeats a whole grouped expression, not just its last symbol.

Precedence and grouping

Order: repetition first, concatenation second, union last.

Read without ambiguity

ab* means a(b*).

L(ab*) = {a, ab, abb, ...}.

(ab)* repeats the block ab.

a | ba means a | (ba).

It describes {a, ba}.

Compare the languages

ab*: a, ab, abb are members.

(ab)*: ε, ab, abab are members.

a is not in (ab)*.

abab is not in ab*.

Parentheses can change the language.

Always check which symbol or group the star applies to.

Build expressions from verbal rules

Let A = (a | b), so A* describes every string over Σ.

Prefix, suffix and substring

Starts with a: aA*

Ends with ab: A*ab

Contains ab: A*abA*

Exactly two symbols: AA

At least two symbols: AAA*

Check the construction

Ends with ab: A*ab

Choose ε for A*: ab.

Choose ba for A*: baab.

aba ends with ba, not ab.

ε cannot end with ab.

A* permits any finite prefix or suffix, including ε.

Build expressions using counts

Translate a counting condition into fixed symbols and flexible gaps.

Exactly one or two b symbols

Exactly one b: a*ba*

Example: aabaa = aa · b · aa

Exactly two b: a*ba*ba*

Example: baba = ε · b · a · b · a

No other b can appear in a* gaps.

Even number of b symbols

Expression: a*(ba*ba*)*

Each repeated block adds two b symbols.

Zero blocks permits ε, a, aa, …

Example: abba = a · (b · ε · b · a)

aba has one b and is excluded.

Not every language is regular; equal ordered blocks $a^n b^n$ are a later example.

Practice and explained answers

Use Σ = {a, b}; A abbreviates (a | b).

Try these questions

1. All strings containing only a.
2. Strings starting and ending with b.
3. Strings containing at least one a.
4. Exactly one b.
5. Does ε belong to (ab)*?

Answers and reasoning

1. a* (zero or more a symbols).
2. b | bA*b (include single b).
3. A*aA* (one required a).
4. a*ba* (one required b).
5. Yes: choose zero repetitions.

Next: more language-building problems before constructing automata.