Lecture 6 — Regular Expressions

CSE205: Theory of Computation

Minseok Jeon

2026 Fall

Motivation: Searching for Patterns

Often we want to find not a fixed word, but every string *fitting a pattern*. The Unix tool `grep` (*global regular expression print*) prints each line of a file matching such a pattern:

Command	**Matches lines**
`grep "ab*c" f.txt`	containing `ac`, `abc`, `abbc`, ...
`grep -E "^[0-9]+$" f.txt`	that are exactly a number
`grep -E "^(a|b)*a$" f.txt`	over $\{a, b\}$ that end in `a`

The same idea drives find/replace in editors, input validation, log analysis, and the lexical analyser of every compiler.

Regular Expressions

A *regular expression* is an algebraic notation that denotes a language.

- Finite automata describe languages *operationally* (by a machine).
- Regular expressions describe languages *declaratively* (by an algebraic formula).

For example, $(\boldsymbol{a} + (\boldsymbol{b} \cdot \boldsymbol{c}))^*$ stands for the language

$$\{\boldsymbol{\varepsilon},\ \boldsymbol{a},\ \boldsymbol{aa},\ \boldsymbol{bc},\ \boldsymbol{aaa},\ \boldsymbol{abc},\ \boldsymbol{bca},\ \boldsymbol{aaaa},\ \boldsymbol{aabc},\ \ldots\}$$

Regular expressions are widely used in practice: text search, pattern matching, and lexical analysis in compilers.

Syntax of Regular Expressions

Definition (Syntax of regular expressions)

Regular expressions over an alphabet Σ are constructed recursively:

1. **(Basis)** $\emptyset$, $\boldsymbol{\varepsilon}$, and each $\boldsymbol{a} \in \Sigma$ are regular expressions.
2. **(Induction)**
   - If $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are regular expressions, then so are $\boldsymbol{R}_1 + \boldsymbol{R}_2$ and $\boldsymbol{R}_1 \cdot \boldsymbol{R}_2$.
   - If $\boldsymbol{R}$ is a regular expression, then so are $\boldsymbol{R}^*$ and $(\boldsymbol{R})$.

Equivalently, in grammar form:

$$\boldsymbol{R} \;\to\; \emptyset \;\mid\; \boldsymbol{\varepsilon} \;\mid\; \boldsymbol{a} \in \Sigma \;\mid\; \boldsymbol{R}_1 + \boldsymbol{R}_2 \;\mid\; \boldsymbol{R}_1 \cdot \boldsymbol{R}_2 \;\mid\; \boldsymbol{R}^* \;\mid\; (\boldsymbol{R})$$

As an abbreviation we also write $\boldsymbol{R}^+$ for $\boldsymbol{R} \cdot \boldsymbol{R}^*$.

Semantics of Regular Expressions

Definition (Semantics of regular expressions)

A regular expression $\boldsymbol{R}$ denotes a set of strings, written $\boldsymbol{L}(\boldsymbol{R})$, defined inductively:

$$\begin{aligned}
\boldsymbol{L}(\emptyset) &= \emptyset \\
\boldsymbol{L}(\boldsymbol{\varepsilon}) &= \{\boldsymbol{\varepsilon}\} \\
\boldsymbol{L}(\boldsymbol{a}) &= \{\boldsymbol{a}\} \qquad (\boldsymbol{a} \in \Sigma) \\
\boldsymbol{L}(\boldsymbol{R}_1 + \boldsymbol{R}_2) &= \boldsymbol{L}(\boldsymbol{R}_1) \cup \boldsymbol{L}(\boldsymbol{R}_2) \\
\boldsymbol{L}(\boldsymbol{R}_1 \cdot \boldsymbol{R}_2) &= \boldsymbol{L}(\boldsymbol{R}_1)\, \boldsymbol{L}(\boldsymbol{R}_2) \\
\boldsymbol{L}(\boldsymbol{R}^*) &= \big(\boldsymbol{L}(\boldsymbol{R})\big)^* \\
\boldsymbol{L}((\boldsymbol{R})) &= \boldsymbol{L}(\boldsymbol{R})
\end{aligned}$$

Hence the abbreviation $\boldsymbol{R}^+ = \boldsymbol{R} \cdot \boldsymbol{R}^*$ denotes the positive closure: $\boldsymbol{L}(\boldsymbol{R}^+) = \big(\boldsymbol{L}(\boldsymbol{R})\big)^+$.

Operator Precedence

To reduce parentheses we fix a precedence among the operators (from highest to lowest):

1. **Star** R^* (and R^+) — binds tightest.
2. **Concatenation** $R_1 \cdot R_2$.
3. **Union** $R_1 + R_2$ — binds loosest.

So, for example,

$$a + b \cdot c^* \quad \text{means} \quad a + \big(b \cdot (c^*)\big),$$

not $(a + b) \cdot c^*$ or $\big((a + b) \cdot c\big)^*$.

Repeated $+$ and $\cdot$ are read left-associatively — for instance $a + b + c$ means $(a + b) + c$ — though this choice is immaterial, since both operators are associative.

The concatenation dot is often omitted: ab means $a \cdot b$.

Example

Compute $\boldsymbol{L}(\boldsymbol{a}^* \cdot (\boldsymbol{a} + \boldsymbol{b}))$.

$$
\begin{aligned}
\boldsymbol{L}(\boldsymbol{a}^* \cdot (\boldsymbol{a} + \boldsymbol{b})) &= \boldsymbol{L}(\boldsymbol{a}^*)\, \boldsymbol{L}(\boldsymbol{a} + \boldsymbol{b}) \\
&= \left(\boldsymbol{L}(\boldsymbol{a})\right)^* \left(\boldsymbol{L}(\boldsymbol{a}) \cup \boldsymbol{L}(\boldsymbol{b})\right) \\
&= \{\boldsymbol{a}\}^* \{\boldsymbol{a}, \boldsymbol{b}\} \\
&= \{\boldsymbol{\varepsilon}, \boldsymbol{a}, \boldsymbol{aa}, \boldsymbol{aaa}, \ldots\} \{\boldsymbol{a}, \boldsymbol{b}\} \\
&= \{\boldsymbol{a},\ \boldsymbol{b},\ \boldsymbol{aa},\ \boldsymbol{ab},\ \boldsymbol{aaa},\ \boldsymbol{aab},\ \ldots\}
\end{aligned}
$$

That is, all strings of $\boldsymbol{a}$'s and $\boldsymbol{b}$'s consisting of zero or more $\boldsymbol{a}$'s followed by a single $\boldsymbol{a}$ or $\boldsymbol{b}$.

Exercises: From Expressions to Languages

Find the languages denoted by the following regular expressions over $\Sigma = \{\boldsymbol{a}, \boldsymbol{b}\}$, and sketch an equivalent finite automaton.

- $(\boldsymbol{a} + \boldsymbol{b})^*$
- $(\boldsymbol{a} + \boldsymbol{b})^*(\boldsymbol{a} + \boldsymbol{b})$
- $(\boldsymbol{a} \cdot \boldsymbol{a})^*(\boldsymbol{b} \cdot \boldsymbol{b})^*\boldsymbol{b}$

Exercises: From Expressions to Languages (Solutions)

- $(\boldsymbol{a} + \boldsymbol{b})^*$
  All strings over $\{\boldsymbol{a}, \boldsymbol{b}\}$ (including $\boldsymbol{\varepsilon}$): Σ^*.
- $(\boldsymbol{a} + \boldsymbol{b})^*(\boldsymbol{a} + \boldsymbol{b})$
  All *nonempty* strings over $\{\boldsymbol{a}, \boldsymbol{b}\}$: Σ^+.
- $(\boldsymbol{a} \cdot \boldsymbol{a})^*(\boldsymbol{b} \cdot \boldsymbol{b})^*\boldsymbol{b}$
  Strings $\boldsymbol{a}^{2\boldsymbol{n}}\boldsymbol{b}^{2\boldsymbol{m}+1}$ with $\boldsymbol{n}, \boldsymbol{m} \geq 0$: an even number of $\boldsymbol{a}$'s followed by an odd number of $\boldsymbol{b}$'s.

Exercises: From Languages to Expressions

Find regular expressions for the following languages over $\{0, 1\}$ (or $\{a, b\}$):

- $L = \{w \in \{0, 1\}^* \mid 0 \text{ and } 1 \text{ alternate in } w\}$
- $L = \{w \in \{0, 1\}^* \mid w \text{ has at least one pair of consecutive zeros}\}$
- $L = \{a^n b^m \mid n \geq 3,\ m \text{ is even}\}$
- $L = \{a^n b^m \mid (n + m) \text{ is even}\}$
- $L = \{a^n b^m \mid n \geq 4,\ m \leq 3\}$

Exercises: From Languages to Expressions (Solutions)

- alternating 0/1:

$$(01)^* + (10)^* + 0(10)^* + 1(01)^* \quad \text{i.e.} \quad (\boldsymbol{\varepsilon} + 1)(01)^*(\boldsymbol{\varepsilon} + 0).$$

- at least one pair of consecutive zeros:

$$(0+1)^*\,00\,(0+1)^*.$$

- $\boldsymbol{n} \geq 3$, $\boldsymbol{m}$ even:

$$\boldsymbol{a\,a\,a\,a}^*\,(\boldsymbol{bb})^*.$$

- $\boldsymbol{n} + \boldsymbol{m}$ even (over $\{\boldsymbol{a}, \boldsymbol{b}\}$, $\boldsymbol{a}^{\boldsymbol{n}}\boldsymbol{b}^{\boldsymbol{m}}$):

$$(\boldsymbol{aa})^*(\boldsymbol{bb})^* + \boldsymbol{a}(\boldsymbol{aa})^*\boldsymbol{b}(\boldsymbol{bb})^*.$$

- $\boldsymbol{n} \geq 4$, $\boldsymbol{m} \leq 3$:

$$\boldsymbol{a\,a\,a\,a\,a}^*\,(\boldsymbol{\varepsilon} + \boldsymbol{b} + \boldsymbol{bb} + \boldsymbol{bbb}).$$

Summary

- A regular expression is an algebraic notation that *denotes* a language $\boldsymbol{L}(\boldsymbol{R})$.
- Its syntax is given inductively from the basis $\emptyset$, $\boldsymbol{\varepsilon}$, $\boldsymbol{a}$ and the operators $+$ (union), $\cdot$ (concatenation), and $*$ (Kleene star).
- Its semantics $\boldsymbol{L}(\boldsymbol{R})$ is defined by structural induction on $\boldsymbol{R}$.
- Precedence: star $>$ concatenation $>$ union.
- **Next:** regular expressions and finite automata denote exactly the same class of languages.