

Introduction to Computer Science Theory

Alvin Lin

August 2017 - December 2017

Regular Languages and Regular Expressions

A **regular expression** over Σ is a description of a language that can be built from $\emptyset, \{\epsilon\}, \{a\}$ for $a \in \Sigma$, using operators, union, concatenation, and Kleene star. For example, the following language over $\{a, b\}$ is described by a regular expression:

$$(\{a\} \cup \{b\})^* \circ \{a\} \circ \{a\} \circ (\{a\} \cup \{b\})^* \cup (\{a\} \cup \{b\})^* \circ \{b\} \circ \{b\} \circ (\{a\} \cup \{b\})^*$$

This can be written a bit more concisely as:

$$(\{a\} \cup \{b\})^* aa(\{a\} \cup \{b\})^* \cup (\{a\} \cup \{b\})^* bb(\{a\} \cup \{b\})^*$$

Example

Given regular expressions to describe:

- The language of strings over $\{a, b\}$ of even length.

$$(aa \cup ab \cup ba \cup bb)^*$$

$$((a \cup b)(a \cup b))^*$$

- The language of strings over $\{a, b, c\}$ in which all s 's precede all b 's and c 's, and all b 's precede all c 's.

$$a^*b^*c^*$$

- The language of strings over $\{0, 1\}$ with length greater than three.

$$(0 \cup 1)(0 \cup 1)(0 \cup 1)(0 \cup 1)(0 \cup 1)^*$$

$$(0 \cup 1)^4(0 \cup 1)^*$$

- The language of strings of odd length of $\{a, b\}$ that contain the substring $\{bb\}$.

$$((a \cup b)^2)^*(abb \cup bba \cup bbb)((a \cup b)^2)^*$$

- The language of strings of $\{0, 1\}$ that do not contain the substring 000.

Is $\{a^i b^i \mid i \in \mathbb{N}\}$?

Equivalence of Finite Automata and Regular Expressions

It is possible to construct a nondeterministic finite automata from a regular expression, and since deterministic finite automata can be constructed from nondeterministic finite automata, the three models can be represented equivalently using all three. This is something we can prove.

Theorem: A language is regular if and only if some regular expressions describes it.

Lemma: If a language is described by a regular expression, then it is regular.

Lemma: If a language is regular, then it is described by a regular language.

Recall the recursive definition of regular languages. R is a regular expression if it is:

1. a , for some $a \in \Sigma$,
2. \emptyset ,
3. ϵ ,
4. $R_1 \cup R_2, R_1 R_2, R_1^*$ and (R_1) , where R_1 and R_2 are regular expressions.

Proof

Base Cases (from the recursive definition):

Case 1: $r = a \in \Sigma$

Case 2: $r = \emptyset$

Case 3: $r = \epsilon$

Induction Hypothesis: Assume that for arbitrary regular expressions r_1, r_2 that nondeterministic finite automata $N_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1), N_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$ exist where $L(r_1) = L(N_1)$ and $L(r_2) = L(N_2)$. Assume without loss of generality $Q_1 \cap Q_2 = \emptyset$.

Case 1: The following NFA $N = (Q, \Sigma, \delta, q_0, F)$ recognizes $L(r_1 \cup r_2)$:

$$\begin{aligned} Q &= Q_1 \cup Q_2 \cup \{q_0\} \text{ where } q_0 \notin Q_1 \cup Q_2 \\ F &= F_1 \cup F_2 \\ \delta : Q \times \Sigma_\epsilon &\rightarrow 2^Q \text{ on } (q, x) \in Q \times \Sigma_\epsilon \\ \delta(q, x) &= \begin{cases} \{q_1, q_2\} & \text{if } (q, z) = (q_0, \epsilon) \\ \delta_i(q, x) & \text{if } q \in Q_i \text{ for } i \in \{1, 2\} \\ \emptyset & \text{otherwise} \end{cases} \end{aligned}$$

You can find all my notes at <http://omgimanerd.tech/notes>. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech