Theory of Computation Slides based on Michael Sipser's Textbook

Moses A. Boudourides[1]

Visiting Associate Professor of Computer Science
Haverford College

[1] Moses.Boudourides@cs.haverford.edu

Sections 0.2 & 1.1

Strings and Languages & Finite Automata

January 14, 2022

Strings, I

Basic Definitions

- An **alphabet** is a nonempty finite set, the elements of which would be called **symbols**. Typically, we will use the Greek letter Σ to denote an alphabet. Examples of alphabets:
  $\Sigma_1 = \{a, b\}, \Sigma_2 = \{0, 1\}, \Sigma_3 = \{a, b, c, \cdots, z\}$, etc.
- A **string over an alphabet** is a finite sequence of symbols from that alphabet, usually written next to one another (i.e., *concatenated*) and not separated by commas. Examples of strings: if $\Sigma_1 = \{a, b\}$, then $abaab$ is a string over Σ_1; if $\Sigma_2 = \{a, b, c, \cdots, z\}$, then $aloha$ is a string over Σ_2.

Strings, II

Basic Definitions, cont.

- For a string x, $|x|$ stands for the **length** (i.e., the number of symbols) of x.
- In addition, for a string x over alphabet Σ and a symbol $\sigma \in \Sigma$,

$$\begin{aligned} n_\sigma(x) = {} & \text{the number of occurrences of the symbol } \sigma \\ & \text{in the string } x. \end{aligned}$$

- The **null string** is a string over Σ, which is defined as the string with zero length and it is denoted by ε, no matter what the alphabet Σ is. As said, $|\varepsilon| = 0$.
- The **set of all strings over alphabet** Σ will be written Σ^*. For the alphabet $\{a, b\}$, we have

$$\{a, b\}^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}.$$

Strings, III

Basic Definitions, cont.

- If string x (over alphabet Σ) has length n, we can write $x = x_1 x_2 \cdots x_n$, where each $x_i \in \Sigma$. The **reverse** of x, written x^R, is the string obtained by writing x in the opposite order, i.e., $x^R = x_n x_{n-1} \cdots x_1$. String x is called **palindrome** if $x = x^R$.
- If we have string x of length m and string y of length n, the **concatenation** of x and y, written xy, is the string obtained by appending y to the end of x, as in $x_1 \cdots x_m y_1 \cdots y_n$.
- If s is a string and $s = xyz$, for three strings x, y and z, x is called **prefix** of s, z **suffix** of s, and y **substring** of s. Strings x, y, z are called **proper prefix–suffix–substring** of s, respectively, if they are different that s.
- The **lexicographic order** of strings is the same as the familiar dictionary order. The **shortlex order** or simply **string order** is a lexicographic order, in which shorter strings precede longer strings. Thus, for example, the string ordering of all strings over the alphabet $\{a, b\}$ is $\{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$.

Strings, IV

Definition of string exponentation

For every string x and integer $k \geq 0$, x^k is a string when defined as:

$$x^k = \begin{cases} \varepsilon, \text{ for } k = 0, \\ x^{k-1}x, \text{ for } k > 0. \end{cases}$$

Operations on Strings

For any strings x, y, z over alphabet Σ, i.e., $x, y, z \in \Sigma^*$,

- $\varepsilon x = x\varepsilon = x$, i.e., ε is the *neutral* or *identity element* of concatenation, considered as a binary relation on Σ^*.
- if either $xy = x$ or $yx = x$, then $y = \varepsilon$,
- $|xy| = |x| + |y|$,
- $(xy)z = x(yz)$, i.e., concatenation is an associative relation and, thus, we may write xyz without specifying how the fractors are grouped.

Languages

Definition

A **language** L over alphabet Σ is a set of strings over Σ, i.e., $L \subseteq \Sigma^*$.

Examples of Languages

- $\varnothing$ is the empty language (since $\{\varnothing\} \subset \Sigma^*$).
- $\{\sigma \mid \sigma \in \Sigma\}$ is the language of all symbols, considered as strings with length 1.
- $\{\varepsilon, a, aab\}$ is a language over $\{a, b\}$ consisting of three strings.
- $Pal(\Sigma)$ is the language of all palindromes over Σ.
- $\{x \in \{a, b\}^* \mid n_a(x) > n_b(x)\}$.
- $\{x \in \{a, b\}^* \mid |x| \geq 2$ and x begins and ends with $b\}$.

Remark

As languages, $\{\varepsilon\} \neq \varnothing$. In addition, $\varepsilon \in \Sigma^*$, though other languages $L \subset \Sigma^*$ may or may not contain ε (in the above examples only the third and the fourth do).

Operations on Languages, I

Propositions on Set Operations and Concatenations of Languages

Let L, L_1, L_2 be languages over Σ. Then:

- $L_1 \cup L_2, L_1 \cap L_2, L_1 \setminus L_2$ and the complement of L, denoted $\overline{L}$ and defined as $\overline{L} = \Sigma^* \setminus L$, are all languages over Σ.
- The **concatenation of two languages** L_1 and L_2, denoted $L_1 \circ L_2$ and defined as $L_1 \circ L_2 = \{xy \mid x \in L_1 \text{ and } y \in L_2\}$, is a language over Σ.
- $L \circ \{\varepsilon\} = \{\varepsilon\} \circ L = L$. (Notice: $L \circ \varnothing = \varnothing \circ L = \varnothing$.)
- If $L \circ L_1 = L$ (or $L_1 \circ L = L$), it is not always true that $L_1 = \{\varepsilon\}$ (a counterexample is given by $L_1 = \Sigma^*$).
- However, if L_1 is a language such that $L \circ L_1 = L$ (or $L_1 \circ L = L$), for *every* language L, then $L_1 = \{\varepsilon\}$.

Operations on Languages, II

Definition of Language Exponentation

For every language L and integer $k \geq 0$, L^k is a language when defined as:

$$L^k = \begin{cases} \{\varepsilon\}, & \text{for } k = 0, \\ L^{k-1} \circ L, & \text{for } k > 0. \end{cases}$$

Remark

$$\Sigma^k = \{x \in \Sigma^* \mid |x| = k\}.$$

Operations on Languages, III

Definition of Language Closures

For every language L, the **Kleene closure** or **Kleene star** of L and the **positive closure** of L are the languages, denoted L^* and L^+, respectively, which are defined by

$$L^* = \bigcup_{k \geq 0} L^k,$$
$$L^+ = \bigcup_{k \geq 1} L^k.$$

In other words, L^* is the set of strings formed by taking any number of strings (possibly none) from L, possibly with repetitions, and concatenating all of them, and L^+ is the same set, when we should take at least one of such strings. Symbolically:

$$L^* = \{x_1 x_2 \dots x_k \mid k \geq 0 \text{ and each } x_i \in L\},$$
$$L^+ = \{x_1 x_2 \dots x_k \mid k \geq 1 \text{ and each } x_i \in L\}.$$

Operations on Languages, IV

Remark

$$\varnothing^* = \{\varepsilon\} \text{ and } \varnothing^+ = \varnothing,$$
$$\{\varepsilon\}^* = \{\varepsilon\} \text{ and } \{\varepsilon\}^+ = \{\varepsilon\}.$$

Proposition

For any language L:

- $L^* = \{\varepsilon\} \cup L^+$,
- $\varepsilon \in L^*$ and $\varepsilon \in L^+ \iff \varepsilon \in L$,
- $L^+ = L \circ L^* = L^* \circ L$,
- $(L^+)^+ = L^+$,
- $(L^*)^* = L^*$.

Operations on Languages, V

Example

For $a \in \Sigma$, consider the language $L = \{a\}$. Then:

$$L^* = \{\varepsilon, a, a^2, a^3, \dots\} = \sum_{k \geq 0} a^k,$$

$$L^+ = \{a, a^2, a^3, \dots\} = \sum_{k \geq 1} a^k.$$

Example: The case $L^* = L^+ = L$

Let $\Sigma = \{0, 1, 2, 3\}$ and $L = \{x \in \Sigma^* \mid n_3(x) = 0\}$. Clearly, $\varepsilon \in L$. We claim that, for all integers $k \geq 1$, $L^k = L$. Apparently, $L^k \subset L$. In addition, if $x \in L$, then, for any integer $k \geq 1$, $x = \varepsilon^{k-1}x$, i.e., $x \in L^k$, which implies that $L \subset L^k$. Therefore, $L^+ = \bigcup_{k \geq 1} L^k = \bigcup_{k \geq 1} L = L$. Moreover, $L^* = \{\varepsilon\} \cup L^+ = \{\varepsilon\} \cup L = L$ (since $\varepsilon \in L$).

Definition of Finite Automata (FA)

Definition: A Finite Automaton

A **finite automaton** (**FA**) is a 5–tuple $(Q, \Sigma, q_0, F, \delta)$, where

- Q is a finite set called the **states**,
- Σ is a finite set called the **alphabet**,
- $q_0 \in Q$ is the **start state**,
- $F \subseteq Q$ is the **set of accept states**, and
- $\delta \colon Q \times \Sigma \to Q$ is the **transition function**,

For any element q of Q and any symbol $\sigma \in \Sigma$, we interpret $\delta(q, \sigma)$ as the state to which the FA moves, when it is in state q and receives the input σ.

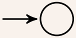
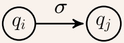
Graphical Representation of FA

Graph Plots of FAs

A FA is drawn as a **labeled directed graph**, in which:

- vertices, drawn as  or 

or 

, correspond to states,
- the start state is drawn as ,
- accept states are drawn as , and
- transition $\delta(q_i, \sigma) = q_j$ is drawn as 

.

Configurations and Yieldings

Definition

Let $M = (Q, \Sigma, q_0, F, \delta)$ be a FA. Any element C of the Cartesian product $Q \times \Sigma^*$ is called **configuration** of M. An **initial configuration** of M is a configuration $C_0 = (q_0, x)$, for $x \in \Sigma^*$, and a **final configuration** of M is a configuration $C_f = (q_f, x)$, for $q_f \in F$ and $x \in \Sigma^*$.

Given two configurations C_i and C_j such that $C_i = (q_i, \sigma y)$ and $C_j = (q_j, y)$, for $q_i, q_j \in Q, y \in \Sigma^*$ and $\sigma \in \Sigma$, we say that configuration C_i **yields in one step** configuration C_j and write

$$C_i \vdash C_j,$$

if

$$q_j = \delta(q_i, \sigma).$$

The Language Accepted by a FA

Definition

Let $M = (Q, \Sigma, q_0, F, \delta)$ be a FA. Given a string $x \in \Sigma^*$, we say that x is **accepted** by M, if there exists a finite sequence of configurations $C_0, C_1, \ldots, C_n$ such that

- $C_0 = (q_0, x), C_n = (q_f, \varepsilon)$, for $q_f \in F$, and
- $C_0 \vdash C_1 \vdash \cdots \vdash C_n$, which is symbolically written as $C_0 \vdash^* C_n$.

The **language accepted** or **recognized** by M is the set

$$L(M) = \{x \in \Sigma^* \mid x \text{ is accepted by } M\}.$$

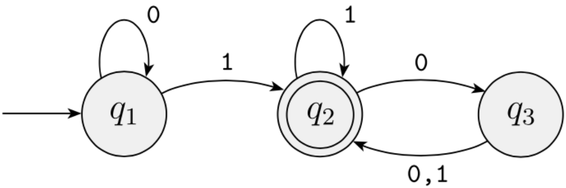
If L is a language over Σ, L is accepted by M if and only if $L = L(M)$.

Definition

A language L over Σ is called **regular language** if there exists a FA $M = (Q, \Sigma, q_0, F, \delta)$ such that $L = L(M)$, i.e., L is accepted (recognized) by M.

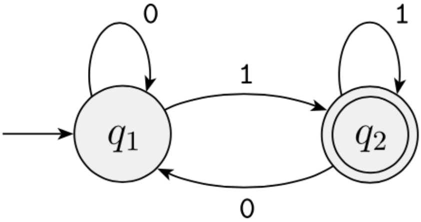
Examples of Finite Automata, I

Example 1:



$L(M) = \{x \mid x \text{ contains at least one 1 and an even number of 0's follow the last 1}\}$

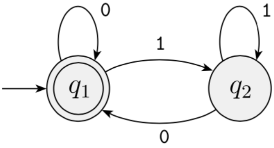
Example 2:



$L(M) = \{x \mid x \text{ ends in } 1\}$

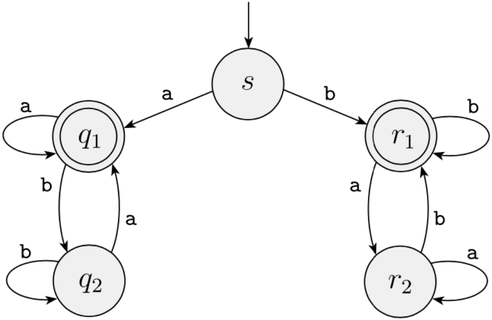
Examples of Finite Automata, II

Example 3:



$$L(M) = \{x \mid x = \varepsilon \text{ or ends in a } 0\}$$

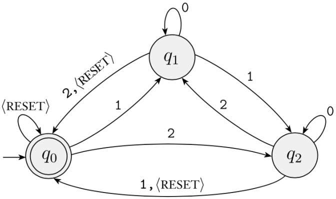
Example 4:



$$L(M) = \{x \mid x \text{ starts and ends with the same symbol}\}$$

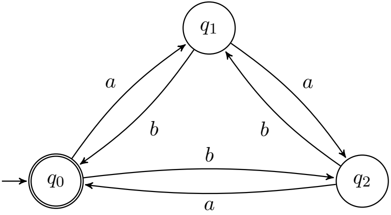
Examples of Finite Automata, III

Example 5:



$$L(M) = \{x \mid x \text{ with sum of symbols equal to } 0 \bmod 3\}$$

Example 6:

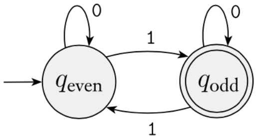


$$L(M) = \{x \mid n_a(x) - n_b(x) = 0 \bmod 3\}$$

Designing FA

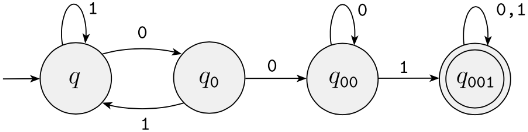
Example 1:

$$L = \{x \in \{0,1\}^* \mid n_1(x) \text{ is odd}\}$$



Example 2:

$$L = \{x \in \{0,1\}^* \mid x \text{ contains the substring } 001\}$$



Regular Operations

Definition

Let L, L_1 and L_2 be languages over the same alphabet Σ. We define three **regular operations** as follows:

- **Union**: $L_1 \cup L_2 = \{x \mid x \in L_1 \text{ or } x \in B\}$.
- **Concatenation**:
  $L_1 \circ L_2 = \{xy \mid x \in L_1 \text{ andr } y \in B\}$.
- **(Kleene) Star**:
  $L^* = \{x_1 x_2 \ldots x_k \mid k \geq 0 \text{ and each } x_i \in L\}$.

Theorem: Closure of Regular Languages under Regular Operations

The class of regular languages is closed under all three regular operations: (i) union, (ii) concatenation, and (iii) Kleene star.