NOTES ON THE THEORY OF COMPUTATION

YANNAN MAO

13th September 2022

Contents

1 Automata and Formal Languages 1

1.1 Finite-State Automata and Regular Languages 1

Automata and Formal Languages

An **alphabet** is a finite set Σ, and a **word over the alphabet** Σ is a finite sequence or **string** of the elements of Σ. If a word w is the sequence $(w_0, \dots, w_n)$ for some $n \in \mathbb{N}$, we may write the word as $w_0 \cdots w_n$. The empty word is denoted by ε. The set of all words over Σ is Σ^*[1]. A **formal language over the alphabet** Σ is a subset of Σ^*.

An **automaton** is an ordered sequence that **accepts** some words over an alphabet. The set of words an automaton accepts forms a language, which is unique, in which case we say the automaton **recognises** the language. Given an automaton M, we may speak of the unique language recognised by M as the **language of the automaton** M and denote it by $L(M)$. A machine may accept no string, in which case the language thereof is $\emptyset$.

1.1 Finite-State Automata and Regular Languages

Definition 1. A **finite-state automaton** is an ordered quintuple $(\Sigma, S, \delta, s_0, F)$ wherein

(a) Σ is an alphabet,

(b) S is a finite set of **states**,

(c) $\delta : S \times \Sigma \to S$ is the **transition function**,

(d) $s_0 \in S$ is the **initial state**, and

(e) $F \subseteq S$ is the set of **final states** or **accept states**.

Let $M = (\Sigma, S, \delta, s_0, F)$ be a finite-state automaton and let $w = w_0 \cdots w_n$ wherein $n \in \mathbb{N}$ be a word over Σ. Then M accepts w if a finite sequence of states $(r_0, \dots, r_n)$ in S exists such that

(a) $r_0 = q_0$,

(b) $\delta(r_i, w_i) = r_{i+1}$ for $i = 0, \dots, n-1$, and

(c) $r_n \in F$.

Definition 2. Let Σ be an alphabet, and let $a \in \Sigma$. Then some $R \subseteq \Sigma^*$ is a **regular language** if

[1] $*$ is the unary operator of Kleene star, defined as $A^* = \{a_0 \cdots a_n : n \in \mathbb{N} \land \forall i \in \mathbb{N}_{<n+1}(a_i \in A)\}$.

(a) $R = \emptyset$,

(b) $R = \{\varepsilon\}$,

(c) $R = \{a\}$,

(d) $R = R_1 \cup R_2$ wherein R_1 and R_2 are regular languages over Σ,

(e) $R = R_1R_2$[2] wherein R_1 and R_2 are regular languages over Σ, or

(f) $R = R_0^*$ wherein R_0 is a regular language over Σ.

An expressed used to characterise a regular language is a **regular expression**.

[2] R_1R_2 is the concatenation of R_1 and R_2, defined as $R_1R_2 = \{xy : x \in R_1 \wedge y \in R_2\}$