Formal Language Theory

Abstract

This is note based on the textbook https://www.cs.utexas.edu/~ear/cs341/automatabook/AutomataTheoryBook.pdf#page=10.08

1 Language and Strings

1.1 Strings

We define the alphabets as Σ, a finite set. We call the item from Σ as symbols or chars.

Definition 1.1 (Strings)**.** We define a string as a finite sequence of symbols drawn from Σ. We define the shortest string formed from Σ as empty string ϵ. The set of all possible strings over Σ is denoted as Σ^* where $*$ is the Kleene star operator [Yifang: defined later].

String functions. Now we introduce some string functions.

For string $s, t \in \Sigma^*$ and symbole $c \in \Sigma$. We use $|\cdot|$ to denote the **length** of s as $|s|$, note that $|\epsilon| = 0$. We use $\#_c(s)$ to denote the **number** of c in s. We use $s||t$ or st to denote **concatenation**. Next, we define string **replication** as s^i for $i \in \mathbb{Z}$. $s^0 = \epsilon$, $s^{i+1} = s^i s$. Lastly, we define string **reversal** as s^R as follows. if $|s| = 0$, then $s^R = s = \epsilon$. For $|s| \geq 1$, there exists $c \in \Sigma$ and $t \in \Sigma^*$ such that $w = ta$. Then, $s^R = at^R$

Fact 1.1 (String Concatenation)**.** *We define strings $s, t, w \in \Sigma^*$. We define some common string concatenation properties as follows.*

- $\forall s, s\epsilon = \epsilon s = s$.
- *concatenation is associative:* $\forall s, t, w, (st)w = s(tw)$.

Now, we can prove the reversal of the concatenation of strings is equivalent to the reversal of each string in reversed concatenation.

Theorem 1.2 (Concatenation and Reversal)**.** *We define $w, t \in \Sigma^*$, then we can have $(wt)^R = t^R w^R$*

Proof. We use math inductions on $|t|$. for $i > 0$, we denote t_i with $|t_i| = i$.

Base case: $|t_0| = 0$.

$$(wt)^R = (w\epsilon)^R = w^R = \epsilon w^R = \epsilon^R w^R = t^R w^R.$$

Induction Step: for $k > 0$, we suppose $(wt_k)^R = t_k^R w^R$
We define a symbol $c \in \Sigma$, then

$$t_{i+1} = t_i c \tag{1}$$

By supposition, we have $(wt_k)^R = t_k^R w^R$. For $k+1$, we can have the following.

$$\begin{aligned}(wt_{k+1})^R &= (wt_k c)^R \\ &= c(wt_k)^R \\ &= c t_k^R w^R \\ &= t_{k+1}^R w^R\end{aligned}$$

where the first step follows from Eq. (1), the second step follows from reversal properties, the third step follows from induction hypothesis, and the last step follows from the reversal of Eq. (1).

Thus, proof is completed by math inductions. □

2 Automaton

2.1 Finite State Automaton

Informally, a finite automaton is a small machine with finite number of states. We feed it one string at a time, and it will update according to the transition function.

Definition 2.1 (DFA)**.** We define a deterministic finite automaton as follows.

$$A = (\Sigma, Q, \delta, q_0, F),$$

where Σ denotes the finite alphabet, Q is the finite set of states, $\delta : Q \times \Sigma \to Q$ is the transition function, $q_0 \in Q$ is the start state, and $F \subseteq Q$ is the accepting states.

The transition function can take the entire input symbols, which we define as follows.

Definition 2.2 ($\widehat{\delta}$ Transition Function)**.** We define a transition functions that takes the entire input (Σ^*) as $\widehat{\delta} : Q \times \Sigma^* \to Q$. We define $\epsilon \notin \Sigma$ as an empty string. For $x \in \Sigma^*$ and $a \in \Sigma$, xa is a appends to x. Then, we can have the following properties.

- $\widehat{\delta}(q, \epsilon) = q$
- $\widehat{\delta}(q, xa) = \delta(\widehat{\delta}(q, x), a)$

2.2 DFA Example

The parity problem is a classic complexity problem, where it signifies whether a model is able to perform modular addition or simulating sign toggling.

Problem 2.1 (n-bit Parity Problem)**.** We define input $x := (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$, then we define

the $\mathsf{Parity}_n : \mathbb{R}^n \to \{0,1\}$ as follows.

$$\mathsf{Parity}_n := \sum_{i=1}^{n} x_i \mod 2$$

Now, we construct a DFA to compute Problem 2.1. We recognize the language $L = \{x \in \{0,1\}^* \mid$ the number of 1s in x is even$\}$. By Definition 2.1, we set the following.

- $\Sigma = \{0,1\}$
- $Q = \{q_{\text{even}}, q_{\text{odd}}\}$
- $q_0 = q_{\text{even}}$
- $F = \{q_{\text{even}}\}$
- $\delta : Q \times \Sigma \to Q$ is as follows:

$$\begin{aligned}
\delta(q_{\text{even}}, 0) &= q_{\text{even}} \\
\delta(q_{\text{even}}, 1) &= q_{\text{odd}} \\
\delta(q_{\text{odd}}, 0) &= q_{\text{even}} \\
\delta(q_{\text{even}}, 1) &= q_{\text{even}}
\end{aligned}$$

References