Lecture 1: Decision Problems and Regular Languages

Date: August 22, 2023.

Decision Problem is a computational problem that expects a Boolean answer on each input.

Given input w. compute $f(w)$ $\Big\}$ $f : \overset{\Sigma^*}{\text{Inputs}} \to \{\top, \bot\}$ $\qquad L_f = \{w \mid f(w) = T\}$ $\qquad L \subseteq \overset{\Sigma^*}{\text{Inputs}}$ $\Big\{$ Given input w. Determine if $w \in L$.

$f_L(w) = \begin{cases} T & \text{if } w \in L \\ \bot & \text{if } w \notin L. \end{cases}$

Inputs encoded as strings over some set of symbols.

Strings:

1. An **alphabet** is a finite set of symbols. For example $\Sigma = \{0, 1\}$, $\Sigma = \{a, b, c, \ldots, z\}$, $\Sigma = \{\langle\text{moveforward}\rangle, \langle\text{moveback}\rangle\}$ are alphabets.
2. A **string/word** over Σ is a finite sequence of symbols over Σ. For example, '0101001', 'string', '$\langle$moveback$\rangle\langle$rotate90$\rangle$'
3. ϵ is the **empty string**.
4. The **length** of a string w (denoted by $|w|$) is the number of symbols in w. For example, $|101| = 3$, $|\epsilon| = 0$, $|\langle\text{moveback}\rangle\langle\text{rotate90}\rangle| = 2$.
5. Σ^* is the set of all strings over Σ; $\Sigma^n = \{w \in \Sigma^* \mid |w| = n\}$
6. **Concatenation** of two strings x and y, denoted either $x \bullet y$ or simply xy, is the unique string containing the symbols of x in order, followed by the symbols in y in order. $x = 10$, $y = 01$ $\quad xy = 1001$ $\quad yx = 0110$
7. y is a **substring** of w if there are strings x, z such that $w = x \bullet y \bullet z$. If $x = \epsilon$ then y is a **prefix** of w. If $z = \epsilon$ then y is a **suffix** of w.

Language over Σ is a set $L \subseteq \Sigma^*$. Examples include $\{\epsilon\}$, $\{w \mid |w| > 5\}$.

- For languages $A, B \subseteq \Sigma^*$, the **concatenation** of A and B is
$$AB = A \bullet B = \{u \bullet v \mid u \in A \text{ and } v \in B\}$$
- For languages $A, B \subseteq \Sigma^*$, their **union** is $A \cup B$, **intersection** is $A \cap B$, and **difference** is $A \setminus B$.
- For $A \subseteq \Sigma^*$, the **complement** of A is $\overline{A} = \Sigma^* \setminus A$.

Powers and Kleene Closure: For a language $L \subseteq \Sigma^*$ and $n \in \mathbb{N}$, define L^n inductively as follows.

$$L^n = \begin{cases} \{\epsilon\} & \text{if } n = 0 \\ L \bullet (L^{n-1}) & \text{if } n > 0 \end{cases}$$

And define $L^* = \cup_{n \geq 0} L^n$, and $L^+ = \cup_{n \geq 1} L^n$.

Alternatively, L^n set of all strings formed by concatenating n strings from L, L^* is the set of all strings formed by concatenating *some* (finite) number of strings from L.

Problem 1. Answer the following questions taking $\Sigma = \{0, 1\}$.

1. What is Σ^0? $\{\epsilon\}$
2. How many elements are there in Σ^3? $|\{000, 001, \ldots \quad 111\}| = 2^3 = 8$.
3. How many elements are there in Σ^n? 2^n
4. For what values of n, is $\Sigma^n \subseteq \Sigma^{n+1}$? Never. $\Sigma^2 = \{00, 01, 10, 11\}$
5. For what values of n, is $\Sigma^n \subseteq \Sigma^*$? Always
6. Let u be an arbitrary string Σ^*. What is $\epsilon \cdot u$? What is $u \cdot \epsilon$? $\epsilon \cdot u = u \cdot \epsilon = u$.

Problem 2. Consider languages over $\Sigma = \{0, 1\}$. L^n | $\forall A \subseteq \Sigma^*$. $\phi \cdot A = \phi = A \cdot \phi$

1. What is $\emptyset^0$? $\{\epsilon\}$ | $\phi^1 = \phi \cdot \phi^0 = \phi \cdot \{\epsilon\} = \phi$
2. Let $L \subseteq \Sigma^*$. What is $|L^*|$? Is it finite? Infinite? [$L = \{\epsilon\}$, $|L^*| = |\{\epsilon\}| = 1$; $\phi^* = \{\epsilon\}$
   In all other cases $|L^*|$ infinite.
3. What is $\emptyset^+$, $\{\epsilon\}^+$? $\phi^+ = \phi$, $\{\epsilon\}^+ = \{\epsilon\}$.

For set A, $|A|$ = # elements in A.

Regular Languages over alphabet Σ are inductively defined as follows.

- $\emptyset$ is a regular language — ϕ — $L(\phi) = \phi$
- $\{\epsilon\}$ is a regular language — ϵ — $L(\epsilon) = \{\epsilon\}$
- $\{a\}$ is a regular language for every $a \in \Sigma$ — a — $L(a) = \{a\}$
- If A, B are regular languages then $A \cup B$ is regular — $(r_1 + r_2)$ — $L((r_1 + r_2)) = L(r_1) \cup L(r_2)$
- If A, B are regular then AB is regular — $(r_1 r_2)$ — $L((r_1 r_2)) = L(r_1) L(r_2)$
- If A is regular then A^* is regular — (r_1^*) — $L((r_1^*)) = (L(r_1))^*$.

Regular Expression

Regular Expression Conventions: To avoid excessive use of parenthesis, the following notational convention will be adopted.

- Precedence order: *, $\cdot$, $+$. For example $r + s^*t$ denotes $(r + ((s^*)t))$
- Associativity: $r + s + t = ((r + s) + t) = (r + (s + t))$ and $rst = ((rs)t) = (r(st))$.

ϕ: Given Input w
Answer no.

Problem 3. Prove the following statements.

1. For any $w \in \Sigma^*$, $L = \{w\}$ is a regular language.
2. For any finite set $L \subseteq \Sigma^*$, L is regular.
3. The set of all strings Σ^* is regular.

$((A B) C) = (A(B C))$

Let $w = a_1 a_2 \ldots a_n$ $(a_i \in \Sigma)$

$\{w\} = \{a_1\}\{a_2\}\{a_3\} \cdots \{a_n\}$. $\leftarrow$

$\rightarrow$ $\forall w.$ $\{w\}$ is regular language. $\Leftrightarrow$ $\forall n$ $\underline{\forall w. |w|=n, \{w\} \text{ is regular language}}$

Prove by induction on $|w|$.

Base Case: $n=0$. $\{\epsilon\}$ is regular language. (by defn)

Ind Hyp: Assume $\forall k<n$, $\forall w.$ $|w|=k$, $\{w\}$ is regular.

Ind Step: $w = a \cdot u$, $|u| = n-1$. and $a \in \Sigma$.

$\{w\} = \{a\}\{u\}$ regular.

regular $\leftarrow$ $\rightarrow$ regular (ind hyp)

$L \subseteq \Sigma^*$ if $|L|$ is finite is regular. by induction on $|L|$.

Σ^* is regular because. Σ is Regular (by 2) and Σ^* is Kleene closure of a regular languages.

Problem 4. Describe the following regular expressions in English.

1. $(0+1)^*$
2. $0\emptyset$
3. $0^* + (0^*10^*10^*10^*)^*$
4. $(0+1)^*001(0+1)^*$
5. $(10)^* + (01)^* + 0(10)^* + 1(01)^*$
6. $(\epsilon+1)(01)^*(\epsilon+0)$
7. $(0+\epsilon)(1+10)^*$

Problem 5. Describe the following languages as a regular expression.

1. All binary strings that have 00 as a substring
2. All binary strings such that the third character from the end is 1
3. All binary strings that have 00 as a substring but do not contain 011 as a substring