CSE331	Automata and Computability

Lecture 2.1: Basic concepts and Terminology-2

Presenter: Azwad Anjum Islam (AAI) *Scribe: Mujtahid Al-Islam Akon (AKO)*

In this lesson, you will learn some more background concepts and terminologies related to the Automata Theory.

Languages and their alphabets

- Recall, a language is just a set of strings. For example-
  - $\{\ldots, -2, -1, 0, 1, 2, \ldots, 5647, 5648, \ldots\}$ is a language of all possible integers – from negative infinity to positive infinity.
    Note: every member of this set are strings e.g. $0, -2, 5647$ etc. might look like single characters or digits or numbers, however, you have to consider them as strings.
  - $\{x, y, xx, xy, yx, yy, \ldots\}$ is a language consisting of all possible strings constructed using the symbols x and y.
  - $\{5318008\}$ is a language consisting of only a single member.
  - $\{\epsilon\}$ is also another language with only a member and that is empty string. So, its *cardinality* = 1.
  - $\{\}$ is a language which belongs nothing. This is called an **empty language**. Thus, its *cardinality* = 0.
    Note: $\{\epsilon\}$ and $\{\}$ might feel similar but they are not. The first one has **a** member whereas the later does not. Also, compare their cardinality.
- Every language has an alphabet of their own. **Recall**, alphabets are the symbols that might be encountered in any string of that language. if you don't remember what alphabets are alphabets are just set of symbols. For example-
  - For Bangla language, there is an infinite number of strings all of which are built from the alphabet set, $\sum =$ {অ, আ, ..., ক, খ, ...}
  - The language of all possible binary strings, $L = \{0, 1, 00, 01, 10, 11, \ldots, 10101110, \ldots\}$ So, its corresponding alphabet set be $\sum = \{0, 1\}$
  - The language of all valid phone numbers , $L = \{01710101010,\ +8801710101010,\ 01710 - 101010, \ldots\}$. So, its corresponding alphabet set be $\sum = \{0, 1, 2, \ldots, 9, -, +\}$
- A language is generally infinite, however, most of the cases, the alphabet sets are finite.

Relationship between a finite state machine and its language

- The following is a generic finite state machine without output-

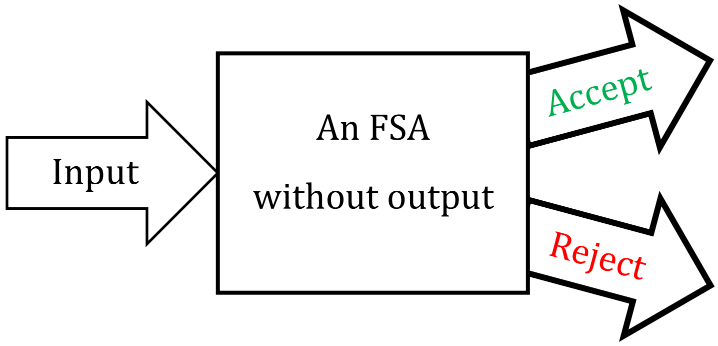

Figure 1: A generic finite state machine without output

- It takes strings as inputs. After processing, some of those inputs gets accepted and some gets rejected. Let's define ***an FSA that accepts positive integers of length*** $\mathbf{2}$ and find out its inputs, its corresponding language and the alphabet set**-**
  - **Define input:** Let the machine is called $\boldsymbol{A}$.Recall that inputs are valid strings that we decide to feed into machine $\boldsymbol{A}$. For the above problem, all the **integers** can be its probable inputs. So, let's assume that only integers numbers can be fed into this machine. Note that integer set is defined as, $Z = \{-\infty, \dots, -2, -1, 0, 1, 2, \dots, \infty\}$. So, $0, 12, 197, 54, 89, 45, -15$ etc. can all be the valid inputs for this machine. however, strings like 123.12 (a decimal fraction), ∀∂℃ℵ∃ℶ or $Dhaka$ are not valid input strings for this machine.
  - **Define alphabet set:** For machine $\boldsymbol{A}$, only the symbols from 0 to 9 and a (+) sign and a (-) sign can be can be used to construct the input. So, the alphabet set for this machine is, $\Sigma = \{1,2,3, \dots, 9, +, -\}$.
  - **Define the language:** Among the valid input strings, the machine will **accept** only those string which are positive and whose length is exactly two, and **reject** the others. So, 12, 54, 89, 45 will be accepted, however, 0, 197 and -15 will be rejected. *The complete set of Inputs accepted by the machine is called the language of that finite state machine* which is denoted by $\boldsymbol{L(A)}$. So, for machine A, the language will be, $L(A) = \{12, 54, 89, 45, \dots\}$.

Find out by yourself: Is $L(A)$ for the above machine finite or infinite?

Different operations on the Language

Languages are sets, so, all the set operations (e.g. union, intersection, complement, etc.) are also applicable on languages.

- **The Universal Language/Set:** The language that contains all possible strings that can be generated using the symbols in the alphabets is called the universal language (denoted by $\boldsymbol{U(L)}$ $\boldsymbol{or}$ $\boldsymbol{U}$). For example-

- For the alphabet, $\sum = \{a, b\}$ of a Language L, the universal language will be $\boldsymbol{U} = \{\epsilon, a, b, aa, ab, ba, bb, aaa, \dots\}$

  Note 1: The empty string, $\boldsymbol{\epsilon}$ is the member of the all the universal sets.

  Note 2: The language, L is the subset of the universal set, $U(L)$.

- **Union of Two Languages:** If a language L_1 has some strings, and another language L_2 has some more strings, then their union language, $L_{union} = L_1 \cup L_2$, will have all the strings that are either present in L_1 or in L_2. For example-
  - If $L_1 = \{a, aa, aaa, \dots.\}$ and $L_2 = \{a, an, the\}$, then their union,
    $$L_{union} = L_1 \cup L_2 = \{a, an, the, aa, aaa, \dots\}$$

  Note: In the above example, Alphabet for L_1 is $\Sigma_1 = \{a\}$ and for L_2 is $\Sigma_1 = \{a, e, h, n, t\}$. So, *alphabet* set of their union set L_{union} will be,
  $$\Sigma_{union} = \Sigma_1 \cup \Sigma_1 = \{a, e, h, n, t\}$$

- **Intersection of Two Languages:** If a language, L_1 has some strings, and another language, L_2 has some more strings, then their intersection language, $L_{intersection} = L_1 \cap L_2$, will have only those strings that are present in both L_1 and L_2.
  - If $L_1 = \{a, aa, aaa, \dots.\}$ and $L_2 = \{a, an, the\}$, then their intersection,
    $$L_{intersection} = L_1 \cap L_2 = \{a\}$$

  Note: In the above example, Alphabet for L_1 is $\Sigma_1 = \{a\}$ and for L_2 is $\Sigma_1 = \{a, e, h, n, t\}$. So, *alphabet* set of their intersection set $L_{intersecton}$ will be,
  $$\Sigma_{intersection} = \Sigma_1 \cap \Sigma_1 = \{a\}$$

- **Complement of a Language:** If a language, L has some strings, its complement language, $\bar{L}$ will contain all the other strings of the universal set $U(L)$, i.e., the strings that are not present in L.
  - If $L = \{\epsilon,\ a, aa\}$ and its alphabet $\Sigma = \{a\}$, then, the universal language,
    $$U(L) = \{\epsilon,\ a, aa, aaa, aaaa, aaaaa, \dots\}$$
    So, the complement language of L be–
    $$\bar{L} = U - L = \{aaa, aaaa, aaaaa, \dots\}$$
    Note: As the same set of symbols is used in both L as well as $\bar{L}$, the alphabet for L and $\bar{L}$ both will be the same. i.e.
    $$\Sigma_{complement} = L = \{a\}$$

- **Concatenation of Two Languages:** Concatenation means joining together the strings from the two languages. It is similar to the concatenation of two sets that is already discussed (refer to lecture-1.2).
  - Let, the language, $A = \{a, bb\}$ and the language $B = \{00, 10, 110\}$, then the concatenation of A and B will be,
    $$\boldsymbol{A} \cdot \boldsymbol{B} = \{a00, a10, a110, bb00, bb10, bb110\}$$
    and the concatenation of B and A will be,
    $$\boldsymbol{B} \cdot \boldsymbol{A} = \{00a, 00bb, 10a, 10bb, 110a, 110bb\}$$
    Note: In the above alphabet for A is $\Sigma_A = \{a, b\}$ and for B is $\Sigma_B = \{0,1\}$. So, the alphabet set of the concatenation of A and B will be,
    $$\Sigma_{A \cdot B} = \Sigma_A \cup \Sigma_B = \{a, b, 0,1\}$$

- **Self-concatenation of a Language:** Self-concatenation means to concatenate a Language with itself.

- Let, D be a set, defined as, $D = \{0,1,2,3,4,5,6,7,8,9\}$ (all single digit numbers). Now, the self-concatenation of D will be, $\boldsymbol{D \cdot D} = \{00,01,02,03,\ldots,10,11,12,13,\ldots,99\}$. $\boldsymbol{D \cdot D}$ is often written as $\boldsymbol{D^2}$.

  Therefore,

$$\begin{aligned} D^2 &= D \cdot D \\ &= \{00,01,02,03,\ldots,10,11,12,13,\ldots,99\} \\ &= \{all\ 2 - digit\ positive\ integers\} \\ D^3 &= D \cdot D \cdot D \\ &= D^2 \cdot D \\ &= \{000,001,002,\ldots,100,101,\ldots,999\} \\ &= \{all\ 3 - digit\ positive\ integers\} \\ D^4 &= D \cdot D \cdot D \cdot D \\ &= D^3 \cdot D \\ &= \{0000,0001,0002,\ldots,1000,\ldots,9999\} \\ &= \{all\ 4 - digit\ positive\ integers\} \\ \therefore D^n &= \{all\ n - digit\ positive\ integers\} \end{aligned}$$

  Note that,

$$\begin{aligned} D^0 &= \{all\ 0 - digit\ positive\ integers\} \\ &= \{\epsilon\} \end{aligned}$$

- If we take the union of all these sets/languages, we get **the Kleene Closure** (denoted by an asterisk,*). So, for the above example,

$$D^* = D^0 \cup D^1 \cup D^2 \cup D^3 \cup \ldots \cup D^\infty$$

- **The Kleene Closure:** Formally, if L is a set/language, the Kleene Closure of L is –

$$L^* = \bigcup_{i=0}^{\infty} L^i = L^0 \cup L^1 \cup L2 \cup L3 \cup \ldots \cup L^\infty$$

  - **Example:** If $L = \{a, bc\}$, then

$$\begin{aligned} L^0 &= \{\epsilon\} \\ L^1 &= \{a, bc\} \\ L^2 &= \{aa, abc, bca, bcbc\} \\ L^3 &= L^1 \cup L^2 \\ &= \{a, bc\} \cup \{aa, abc, bca, bcbc\} \\ &= \{aaa, aabc, abca, abcbc, bcaa, bcabc, bcbca, bcbcbc\} \\ \therefore L^* &= L^0 \cup L^1 \cup L2 \cup L3 \cup \ldots \cup L^\infty \\ &= \{\epsilon\} \cup \{a, bc\} \cup \{aa, abc, bca, bcbc\} \cup \ldots \\ &= \{\epsilon, a, bc, aa, abc, bca, bcbc, aaa, aabc, abca, abcbc, bcaa, bcabc, \ldots\} \end{aligned}$$

- **The Positive Closure:** It is defined by –

$$L^+ = \bigcup_{i=1}^{\infty} L^i = L^1 \cup L^2 \cup L^3 \cup \ldots \cup L^\infty$$

  Note: The only difference between L^+ and L^* is that L^* has an extra element i.e. $L^0 = \{\epsilon\}$. So, we can write, $L^+ = L^* - \{\epsilon\}$

  - **Example:** From the above example, as $L = \{a, bc\}$, then

$$L^* = \{\epsilon, a, bc, aa, abc, bca, bcbc, aaa, aabc, abca, abcbc, bcaa, bcabc, \ldots\}$$

    Therefore,

$$\begin{aligned} L^+ &= L^* - \{\epsilon\} \\ &= \{a, bc, aa, abc, bca, bcbc, aaa, aabc, abca, abcbc, bcaa, bcabc, \ldots\} \end{aligned}$$

- **Interpretation of Σ***: Assume that Σ is an alphabet set for a language L. Now, we get,
  Σ^0 = $\{\epsilon\}$
  Σ^1 = the set of the symbols themselves = Σ
  Σ^2 = the set of all strings of length 2 that can be created using the alphabet, Σ.
  Σ^3 = the set of all strings of length 3 that can be created using the alphabet, Σ.
  $\vdots$
  Σ^n = the set of all strings of length n that can be created using the alphabet, Σ.
  So, $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^1 \cup \ldots \cup \Sigma^\infty$ is the language that will contain all possible strings of any length that can be generated using Σ, i.e. the universal set of L.

Summary

- Refer to figure 1. In a finite state machine, M –
  - It has an alphabet set = Σ
  - The input set for M = all the strings generated by from the symbols of Σ = the universal set = $\boldsymbol{\Sigma}^*$
  - The set of accepted strings = $\boldsymbol{L(M)}$
  - The set of strings that the machine rejects = $\overline{\boldsymbol{L(M)}}$
- **Example:** Consider the language that contains all valid JAVA identifiers. A java identifier must follow the following rules-
  (a) Identifiers may contain lower-case and upper-case **letters**, **digits** from 0 to 9, the dollar sign ($), and the underscore (_)
  (b) Identifiers must be of **length one or more**
  (c) Identifiers must **not start with a digit**.

  Let's find out its alphabet and language.
  - **Alphabet set:** According **to rule-(a**), the alphabet set, Σ consists of-
    - The set of letters, $L = \{a, b, c, \ldots, z, A, B, \ldots, Z\}$
    - The set of digits, $D = \{0, 1, 2, \ldots, 9\}$
    - The set of other two symbols, $S = \{\$, \_\}$

    If we combine all of the above, we shall get the alphabet set. Therefore, the alphabet set,
    $$\Sigma = L \cup D \cup S = \{a, b, \ldots, 0, 1, \ldots, \$, \_\}$$
  - **Language set:** According to **rule-(b),** if we take the positive closure of Σ, we get the language that contains all possible strings of length ≥ 1 i.e.
    $$\Sigma^+ = (L \cup D \cup S)^+$$
    Note: The Kleene Closure contains ϵ, which is not a valid identifier. So, we consider the Positive Closure here.
    However, according to **rule-(c),** the strings cannot start with a digit i.e. the symbols from the set, D defined above. So, we concatenate $(L \cup S)$ before Σ^+ to force it to start with either a letter or a symbol. Now we get,
    $$(L \cup S)\Sigma^+ = (L \cup S)(L \cup D \cup S)^+$$
    Now notice that $(L \cup S)$ has a length 1 and $(L \cup D \cup S)^+$ has minimum length 1 too. So, $(L \cup S)(L \cup D \cup S)^+$ must have a length of at least two i.e. the above expression will fail to generate single length identifiers like $x, a, \$$ etc. So, to accommodate those, we need to change the positive closure to the Kleene closure which can be ϵ allowing the expression's minimum length to be 1. So, our final language be,

$$\boldsymbol{L}(\boldsymbol{M}) = (\boldsymbol{L} \cup \boldsymbol{S})\boldsymbol{\Sigma}^{*} = (\boldsymbol{L} \cup \boldsymbol{S})(\boldsymbol{L} \cup \boldsymbol{D} \cup \boldsymbol{S})^{*}$$

For example, to construct a single length identifier, $\boldsymbol{x}$ –

- $(L \cup S)$ will produce x
- $(L \cup D \cup S)^*$ will produce ϵ.

So, the $L(M)$ will produce $x \cdot \epsilon = x$.
