Formal Languages

Lecture 3

October 11, 2021

Objectives

By the end of this lecture, you should be able to

- Identify alphabets, strings, and languages.
- Identify prefixes, suffixes, and substrings of a given string.
- Prove simple properties of languages.

Alphabets

- An **alphabet** is a nonempty finite set of symbols.
  - A symbol is a physical entity that we shall not formally define; we shall rely on intuition.
- Alphabets are typically denoted by the uppercase Greek letters Σ and Γ.
- **Examples of alphabets:**
  - $\Sigma_1 = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}, \mathtt{g}, \mathtt{h}, \mathtt{i}, \mathtt{j}\}$.
  - $\Sigma_2 = \{\mathtt{0}, \mathtt{1}\}$.
  - $\Gamma = \{\$, \#, @, ©, \S, \P\}$

Notational Issues

Let Σ be an alphabet.

- Recall that
  - Σ^2 is the set of pairs of symbols from Σ.
  - Σ^3 is the set of triples of symbols from Σ.
  - Σ^k is the set of k-tuples of symbols from Σ.
- We shall drop the $\times$ in Cartesian products:
  - $\Sigma^m \times \Sigma^n = \Sigma^m \Sigma^n$

Strings

- A **string over an alphabet** is a k-tuple (i.e., finite sequence) of symbols from the alphabet.
- Symbols of a string are usually written next to one another, and not using the standard tuple-notation.
- **Example**
  - Let $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ be an alphabet.
  - Possible strings over Σ include: `a, b, aa, bb, ab, ba, aaa, bbb, aba,` etc.
- The **length** of a string w, denoted $|w|$, is the number of symbols it contains.
  - If $w \in \Sigma^k$, then $|w| = k$.

The Empty String

For any alphabet Σ

- Define $\Sigma^0 = \{\varepsilon\}$.
  - ε denotes the **empty string**—the string consisting of no symbols taken from Σ.
  - Note that $|\varepsilon| = 0$.
  - *Warning:* do not confuse the empty string ε with the empty space " ".
- Define $\Sigma^+ = \Sigma \cup \Sigma^2 \cup \cdots = \bigcup_{n=1}^{\infty} \Sigma^n$.
- Define $\Sigma^* = \Sigma^0 \cup \Sigma^+ = \bigcup_{n=0}^{\infty} \Sigma^n$.
  - Note that $\varepsilon \in \Sigma^*$ and $\varepsilon \notin \Sigma^+$.

Concatenation

- Let w and v be strings in Σ^*.
- The **concatenation** of w and v, written wv is the string in Σ^* resulting from appending v to the end of w.
  - Note that concatenation is associative but not commutative.
- **Example**: Let $\Sigma = \{\mathtt{a}, \mathtt{b}\}$, $w = \mathtt{ab}$, $v = \mathtt{bab}$, and $u = \mathtt{bba}$.
  - $wv = \mathtt{abbab} \neq vw = \mathtt{babab}$.
  - $w(vu) = \mathtt{ab}(\mathtt{babbba}) = \mathtt{abbabbba} = (wv)u = (\mathtt{abbab})\mathtt{bba} = \mathtt{abbabbba}$.
- Note:
  - $\varepsilon w = w\varepsilon = w$, for any $w \in \Sigma^*$.
  - $|wv| = |w| + |v|$.
  - $\overbrace{ww\cdots w}^{k} = w^k$.

Prefixes

- If $u, v \in \Sigma^*$, and $w = uv$, then u is a **prefix** of w.
  - If $v \neq \varepsilon$, then u is a **proper prefix** of w.
- **Example**: Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ and consider $w = \mathtt{abbcc}$.
  - The set of prefixes of w is $\{\varepsilon, \mathtt{a}, \mathtt{ab}, \mathtt{abb}, \mathtt{abbc}, \mathtt{abbcc}\}$.
  - Except for `abbcc`, all of the above are proper prefixes of w.
- How many prefixes does a string have? How many are proper?

Suffixes

- If $u, v \in \Sigma^*$, and $w = uv$, then v is a **suffix** of w.
  - If $u \neq \varepsilon$, then v is a **proper suffix** of w.
- **Example**: Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ and consider $w = \mathtt{abbcc}$.
  - The set of suffixes of w is $\{\varepsilon, \mathtt{c}, \mathtt{cc}, \mathtt{bcc}, \mathtt{bbcc}, \mathtt{abbcc}\}$
  - Except for abbcc, all of the above are proper suffixes of w.
- How many suffixes does a string have? How many are proper?

Reverse

- The **reverse** of w, denoted $w^{\mathcal{R}}$, is the string obtained by writing the symbols of w in the opposite order.

  1. $\varepsilon^{\mathcal{R}} = \varepsilon$
  2. $(au)^{\mathcal{R}} = u^{\mathcal{R}}a$, for $a \in \Sigma$ and $u \in \Sigma^*$.

  – What is the relation between the prefixes/suffixes of w and those of $w^{\mathcal{R}}$?

Substrings

- If $u, v, x \in \Sigma^*$, and $w = uvx$, then v is a **substring** of w.
  - If at least one of u and x is different from ε, then v is a **proper substring** of w.
- **Example**: Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ and consider $w = \mathtt{abbcc}$.
  - The set of substrings of w is $\{\varepsilon, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{ab}, \mathtt{bb}, \mathtt{bc}, \mathtt{cc}, \mathtt{abb}, \mathtt{bbc}, \mathtt{bcc}, \mathtt{abbc}, \mathtt{bbcc}, \mathtt{abbcc}\}$
  - Except for `abbcc`, all of the above are proper substrings of w.
- How many substrings does a string have?
- What is the relation between the substrings of w and those of $w^{\mathcal{R}}$?
- What is the relation between substrings of w, and its prefixes and suffixes?

Languages

- For any alphabet Σ, any subset of Σ^* is called a **language** over Σ.
- Note that
  - A language is a set of strings; an alphabet is a set of symbols. (Although a set of strings of length one would look like an alphabet.)
  - A language may be an empty set (called the **empty language**); an alphabet is nonempty by definition.
  - A language may be infinite; an alphabet is finite by definition.

Examples

- The set $L_1 = \{\varepsilon\}$ is a language over any alphabet. Note that L_1 is not the empty language.
- The set $L_2 = \{\mathtt{0}^n\mathtt{1}^n | n \in \mathcal{Z}, n \geq 0\}$ is the language over $\{\mathtt{0}, \mathtt{1}\}$ consisting of strings starting with zero or more `0`s and followed by the same number of `1`s.
- The set $L_3 = \{w | w \in \Sigma^* \text{ and } w = w^{\mathcal{R}}\}$ is the language over some alphabet Σ, consisting of all **palindromes** in Σ^*.
- The set $L_4 = \{w | w \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^* \text{ and } \mathtt{bab} \text{ is a substring of } w\}$ is a language over $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.
- The set $L_5 = \{w | w \in \{\mathtt{0}, \mathtt{1}, \mathtt{2}\}^* \text{ and } \mathtt{bab} \text{ is a substring of } w\}$ is the empty language.
- The set of executable Java programs is a language over the Java alphabet.

Operations on Languages

- All set operations: union, intersection, difference.
- The concatenation of two languages, L_1 and L_2, is the language

$$L_1 \circ L_2 = \{uv | u \in L_1 \text{ and } v \in L_2\}$$

  - **Example**: Do it yourself.
- Note
  - $L \circ L = L^2$.
  - $\overbrace{L \circ L \cdots \circ L}^{k} = L^k$.
  - $L^0 = \{\varepsilon\}$.
  - $L^+ = L \cup L^2 \cup \cdots = \bigcup_{n=1}^{\infty} L^n$.
  - $L^* = L^0 \cup L^+ = \bigcup_{n=0}^{\infty} L^n$. ($L^*$ is called the **Kleene closure** of L.)

Proving Theorems about Languages

Theorem Let Σ be an alphabet, with languages $L_1, L_2 \subseteq \Sigma^*$. If $L_1 \subseteq L_2$, then ${L_1}^n \subseteq {L_2}^n$, for all $n \in \mathcal{N}$.

Proof. Do it by induction on n.

Proof by Induction

Basis ($n = 1$). If $L_1 \subseteq L_2$, then $L_1^1 = L_1 \subseteq L_2 = L_2^1$.

Induction Hypothesis. If $L_1 \subseteq L_2$, then ${L_1}^k \subseteq {L_2}^k$, for some $k \in \mathcal{N}$.

Induction Step. Suppose that $L_1 \subseteq L_2$. We need to show that ${L_1}^{k+1} \subseteq {L_2}^{k+1}$. Let $w \in {L_1}^{k+1}$. Hence, $w = uv$ where $u \in L_1^k$ and $v \in L_1$. Since $L_1 \subseteq L_2$, then $v \in L_2$ and, by the induction hypothesis, $u \in L_2^k$. Thus, $w = uv \in L_2^k \circ L_2 = L_2^{k+1}$. It follows that ${L_1}^{k+1} \subseteq {L_2}^{k+1}$.

Another One . . .

Example

- Let $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.
- Consider the language L over Σ defined recursively as follows.
  - $\varepsilon \in L$
  - If $w \in L$, then $\mathtt{a}w\mathtt{a} \in L$.
  - If $w \in L$, then $\mathtt{b}w\mathtt{b} \in L$.
  - Nothing else is in L.
- Prove that for all $w \in L$, $|w|$ is even.

Structural Induction

Basis. *(Prove it for the base case(s) of the recursion.)*

Since $|\varepsilon| = 0$, then $|\varepsilon|$ is even.

Induction Hypothesis. *(Assume it is true for some string in L.)*

$|w|$ is even for some $w \in L$.

Induction Step. *(Show that it is true for strings constructed from w according to the recursive rules.)*

Consider the string $\mathtt{a}w\mathtt{a}$. By definition of string concatenation, $|\mathtt{a}w\mathtt{a}| = |\mathtt{a}| + |w| + |\mathtt{a}| = |w| + 2$. By the induction hypothesis, $|\mathtt{a}w\mathtt{a}| = |w| + 2$ is an even number. Similarly, for the string $\mathtt{b}w\mathtt{b}$.

Thus, for all $w \in L$, $|w|$ is even.

Points to take home

- Alphabets.
- Strings.
- Concatenation.
- Substrings.
- Languages.
- Operations on languages.

Next time

- Deterministic Finite Automata.