
بسم الله الرحمن الرحیم

Formal Languages & Automata

Ali Shakiba
Vali-e-Asr University of Rafsanjan
<ali.shakiba@vru.ac.ir>

More Basic Terms

- If w is a string, then w^n is the string obtained by repeating w for n times.
  - Special case: $w^0 = \lambda$ for all w.

- If Σ is an alphabet, then Σ^* is the set of strings obtained by concatenating zero or more symbols from Σ.
  - Σ^* always contains λ
  - $\Sigma^+ = \Sigma^* - \{\lambda\}$ is the set of all nonempty strings

The $*$ operator is known as the Kleene star.

- Even though Σ is finite, Σ^* and Σ^+ are infinite since there is no limit on the string lengths.

We Have Languages!

- Recall that a language has rules that determine whether a given string is a sentence in the language.
  - We often refer to strings in a language as sentences.
- Therefore, any subset of Σ^* is a language.
- The rules are those that determine membership in the subset.

Example Languages

- Let $\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$
- The subset $\{a, aa, aab\}$ is a language on Σ.
    - The membership rule is trivial since we explicitly listed the members of the subset.
- It is a finite language because it contains a finite number of sentences.

Example Languages, *cont'd*

- Let $\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}$
- The subset

$$L = \{a^n b^n : n \geq 0\}$$

  is a language on Σ.
  - The strings $aabb$ and $aaaabbbb$ are sentences in L.
  - The string abb is not a sentence in L.
- This is an infinite language.
  - Most interesting languages are infinite.

String Concatenation

- If you concatenate two sentences of a language, is the result always a sentence of the language?
  - Let $L = \{an : n \text{ is odd}\}$
  - Consider strings u and v in L.
  - Is uv in L?

Languages are Sets

- We can calculate the union, intersection, and difference of two languages.
  - Recall Georg Cantor.

- The complement of language L is

$$\overline{L} = \Sigma^* - L$$

- The reverse of a language is the set of all string reversals:

$$L^R = \{w^R : w \text{ in } L\}$$

Languages are Sets, *cont'd*

- The concatenation of two languages L_1 and L_2:

$$L_1L_2 = \{xy : x \text{ in } L_1, y \text{ in } L_2\}$$

- L^n is L concatenated with itself n times.
  - $L^0 = \{\lambda\}$ and $L^1 = L$

- Star closure

$$L^* = L^0 \cup L^1 \cup L^2 \ldots$$

- Positive closure

Infinite sets!

$$L^+ = L^1 \cup L^2 \ldots$$

Example Languages, *cont'd*

- Let

$$L = \{a^n b^n : n \geq 0\}$$

- Then

$$L^2 = \{a^n b^n a^m b^m : n \geq 0, m \geq 0\}$$

  - n and m are unrelated.
  - The string $aabbaaabbb$ is in L.

- The reverse

$$L^R = \{b^n a^n : n \geq 0\}$$

Rules of a Language

- The rules of a language must enable us to decide, in a finite amount of time, whether a given string is a sentence of the language.

- Two kinds of rules:

  1. Rules that tell us whether or not a string is a sentence of the language.

  2. Rules that tell us how to generate all the sentences of the language.

Example Languages, *cont'd*

- Let $L_1 = \{aa, b\}$
- Then

L_1^*= {λ plus any string composed of factors of aa and b}
= {λ plus all strings of a's and b's where the a's are in even groups

Example Languages, *cont'd*

- Let $L_2 = \{a, ab\}$
- Then

  L_2^* = {λ plus any string composed of factors of a and ab}
  = {λ plus all strings of a's and b's except those that start with b and those that contain any double b}

- Is the string $abaab$ a sentence of L_2^*?
  - Yes. We can factor the string as $(ab)(a)(ab)$ and each factor is in L_2.
  - This factoring is unique.

Example Languages, *cont'd*

- Let $L_3 = \{aa, aaa\}$
- Then

$$L_3^* = \{\lambda \text{ plus any string composed of more than one } a\}$$

- Is the string $a^7 = aaaaaaa$ a sentence of L_3^*?
  - Yes. We can factor the string as $(aa)(aa)(aaa)$ or $(aa)(aaa)(aa)$ or $(aaa)(aa)(aa)$.
  - This factoring is not unique.

What is L^{**}?

- Every string in L^{**} is composed of factors from L^*.
- Every string in L^* is composed of factors from L.
- Therefore, every string in L^{**} is composed of factors from L.
- Therefore, every string in L^{**} is also a string in L^* :

$$L^{**} \subseteq L^*$$

- For any set S, $S \subseteq S^*$. So let $S = L^*$. Then

$$L^* \subseteq L^{**}$$

- Therefore, $L^{**} = L^*$.

Grammars

- The grammar of a language is the set of rules that determine whether or not a sentence is in the language.

- Example: From English grammar:

  <sentence> → *<noun phrase> <predicate>*

  <noun phrase> → *<article> <noun>*

  <predicate> → *<verb>*

  - If *<article>* is "a" or "the", *<noun>* is "boy" or "dog", and *<verb>* is "runs" or "walks", then proper sentences are "a boy runs" and "the dog walks".

Grammars, *cont'd*

- A grammar G is defined as the quadruple

$$G = (V, T, S, P)$$

  where:
  - V is a finite set of objects called variables
  - T is a finite set of objects called terminal symbols
  - S in V is a special symbol called the start variable
  - P is a finite set of production rules

Production Rules

- The production rules specify how the grammar transforms one string to another.
- They define a language associated with the grammar.
- All production rules are of the form $x \rightarrow y$ where $x \in (V \cup T)^{+}$ and $y \in (V \cup T)^{*}$.

Derivations

- If we apply to the string

$$w = uxv$$

  the production rule

$$x \to y$$

  we obtain the new string

$$z = uyv$$

- We say that w derives z, or z is derived from w and write

$$w \Rightarrow z$$

Derivations, *cont'd*

- We can derive successive strings by applying production rules of the grammar in any order.

- We can use a production rule whenever it is applicable, and we can apply it as often as we desire.

- If

$$w_1 \Rightarrow w_2 \Rightarrow \ldots \Rightarrow w_n$$

  then w_1 derives w_n, and we can write

$$w_1 \overset{*}{\Rightarrow} w_n$$

Language Generation by a Grammar

- By applying the production rules in a different order, a grammar can generate many strings.
- The set of all such terminal strings is the language generated by the grammar.
  - AKA the language defined by the grammar.
- Let $G = (V, T, S, P)$ be a grammar. Then the set

$$L(G) = \{w \in T^* : S \overset{*}{\Rightarrow} w\}$$

is the language generated by G.

Sentential Forms

- If $w \in L(G)$, then the sequence

$$S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \ldots \Rightarrow w_n \Rightarrow w$$

  is a derivation of the sentence w.

- The strings $S, w_1, w_2, \ldots, w_n$ are sentential forms of the derivation.

- Sentential forms can contain both variables and terminals.

Grammar Examples

- Let grammar $G = (\{S\}, \{a, b\}, S, P)$
  with P given by

$$S \to aSb$$

$$S \to \lambda$$

- Then

sentential forms

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb$$

  so we can write

$$S \overset{*}{\Rightarrow} aabb$$

- Therefore, the string $aabb$ is in the language

$$L(G) = \{a^n b^n : n \geq 0\}$$

Grammar Examples, *cont'd*

- Find a grammar that generates the language

$$L = \{a^n b^{n+1} : n \geq 0\}$$

- The language is similar to the previous example but with an extra b. We add the production rule

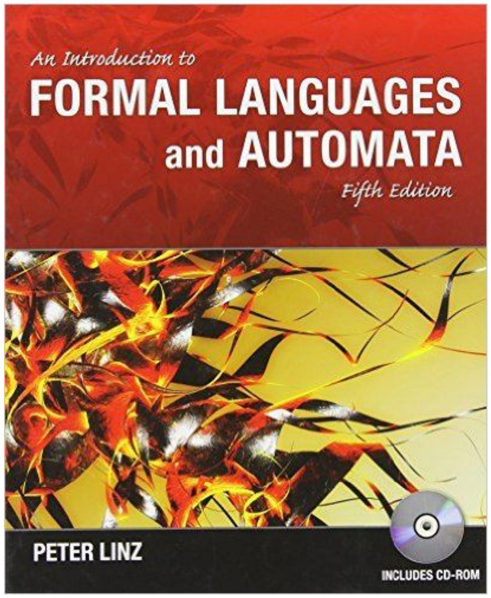
$$S \rightarrow Ab$$

- Therefore, $G = (\{S, A\}, \{a, b\}, S, P)$ with the production rules

$$S \rightarrow Ab$$
$$A \rightarrow aAb$$
$$A \rightarrow \lambda$$

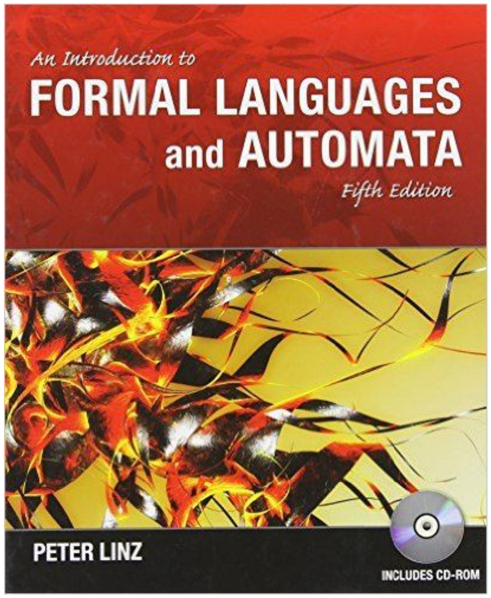
We have learnt …

- Chapter 1



Next session, we will learn …

- Sections 2.1 to 2.2
  - (Non)deterministic Finite Automata



Do not forget ...

- Install JFLAP on your machine
- Solve the first set of assignments
- Due, next Tuesday!