San José State University
Department of Computer Science

Ahmad Yazdankhah
ahmad.yazdankhah@sjsu.edu
www.cs.sjsu.edu/~yazdankhah

Formal Languages

(Part 1)

Lecture 04

Day 04/31

CS 154

Formal Languages and Computability

Spring 2018

Agenda of Day 04

- Announcement
- Summary of Lecture 03
- Lecture 04: Teaching ...
  - Formal Languages (Part 1)

Announcement

- Our first quiz will be this Thursday!
  - Some of the questions are multiple choice.
  - So, please have Scantron 882 E.
  - If you forget, no problem at all! I'll sell it at:

$~~2~~0 Each **Just $19.95!**

- My office hours are changed:

Tuesdays and Thursdays, 7:15pm – 9:15pm

Summary of Lecture 03: We learned ...

Cartesian Products

- In many cases, we need ordered collections like $(x_1, x_2, \ldots, x_n)$.
- Cartesian product can produce ordered collections.
- The Cartesian product of A and B is ...
  - ... the set of all ordered pairs (a , b), where $a \in A \wedge b \in B$.

$$A \times B = \{(a, b) : a \in A \wedge b \in B\}$$

- Does Cartesian product have commutative property?
  - In general, no, but in the following special cases, yes:
  - $(A = B) \vee (A = \phi) \vee (B = \phi)$

- We can extend the Cartesian product to 3 sets:

$$A \times B \times C = \{(a, b, c) : a \in A, b \in B, c \in C\}$$

- Does Cartesian product have associative property?
- No, because:

$$(A \times B) \times C = \{((a, b), c) : a \in A, b \in B, c \in C\}$$

$$A \times (B \times C) = \{(a, (b, c)) : a \in A, b \in B, c \in C\}$$

Any question?

Summary of Lecture 03: We learned ...

Functions

- A function from D to R is ...
  - ... a rule that assigns to some elements of D (domain) a unique element of R (range).
- A total function is ...
  - ... a function that all of its domain elements are defined.
- A partial function is ...
  - ... a function that at least one member of its domain elements is "undefined".

Any question?

Summary of Lecture 03: We learned ...

Graphs

- A graph is a mathematical construct consisting of two sets:
  - A non-empty and finite set of vertices $V = \{v_1 , v_2 , \ldots , v_n\}$
  - A finite set of edges $E = \{e_1 , e_2 , \ldots , e_m\}$
- A walk is ...
  - ... a sequence of edges from v_i to v_n. $(v_i , v_j) , (v_j , v_k) , \ldots , (v_m , v_n)$
- The length of a walk is ...
  - ... the number of edges traversed.
- A path is ...
  - ... a walk that no edge is repeated.
- A simple path is ...
  - ... a path that no vertex is repeated.
- A loop is ...
  - ... an edge from a vertex to itself.
- A cycle is ...
  - ... a path from a vertex (called base) to itself.
- A simple cycle is ...
  - ... a cycle that no vertex other than base is repeated.

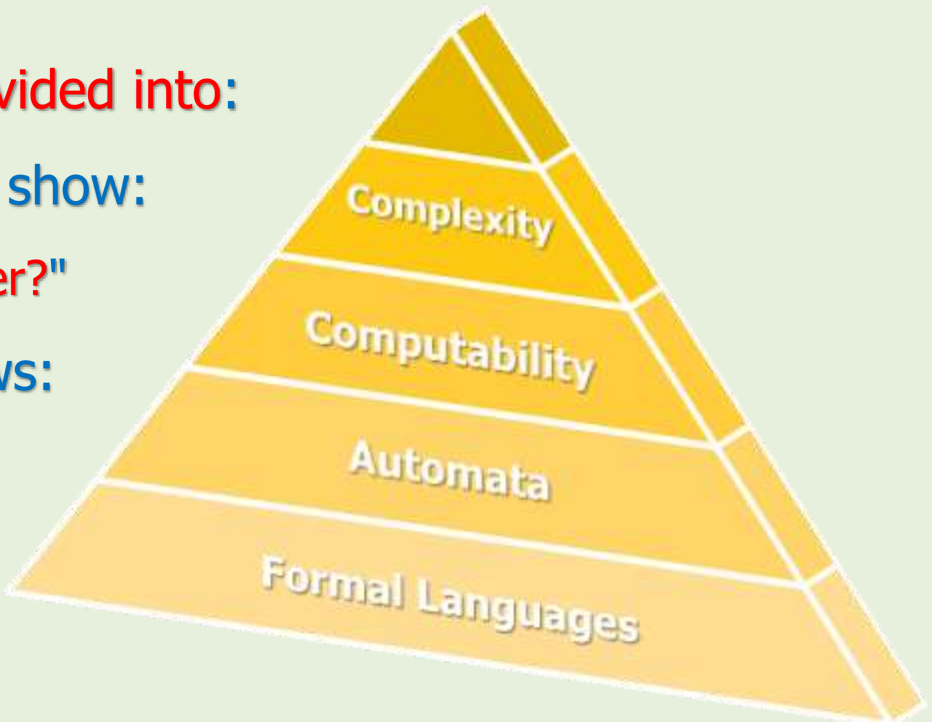
Any question?

Objective of This Lecture

- To review alphabets
- To review strings
- To introduce formal languages
- To examine some languages

Introduction: The Big Picture of the Course

- The foundation of the computer science is the "theory of computation".
- The theory of computation is divided into:
- The first three from the bottom show:
  - "What can be done with computer?"
- The forth one, complexity, shows:
  - "What can be done in practice?"



- Let's start with "Formal Languages"!
- First, we need to have a common understanding about "alphabet" and "strings".

Alphabets & Strings

Alphabets

Definition

- A "nonempty" and "finite" set of "symbols"
- An alphabet is denoted by Σ.

- Note that when we say "an alphabet", we mean a set of symbols.

- We usually (in this course) use lowercase letters a, b, c, ... for alphabets' symbols.
- In some cases, we use digits like 0, and 1 or other symbols as alphabets.

Alphabets Example

Example 1

- Σ = {a, b}   This is our celebrity alphabet!
- Σ = {0, 1}
- Σ = {ε, α, β}

- Can the following set be an alphabet?   Σ = {ЯЕ , ДК , Ѭ , پ , Γ }

Strings

Definition

- A "finite" sequence of symbols from the alphabet

Example 2

- Let $\Sigma = \{a, c, d, e, g, h, l, o, p, r, s, t, u\}$.
- Are these valid strings?
- cat , dog , horse , house , apple

Strings Examples

Example 3

- Let Σ = {a, b}.
- Are the following strings valid strings?
- baba , aabb , bbbbbbbbbbba , ...
- Not all of them!
  - "..." is not a valid string because "." is not in the alphabet!
  - Note that in "formal languages" arena, we don't care whether the strings are meaningful or not!

- We use lowercase letters w, u, v, ... for string variables.
- w = baba
- u = bbbbbbbbbbba

Strings Size (aka Length)

Definition

- The number of symbols in the string
- The size of string w is denoted by |w|.

Example 4

- |aaa| = 3
- |babba| = 5
- |aaba| = 4

- In general:

$$|a_1\ a_2\ \ldots\ a_n| = n$$

Empty String

Definition

- A string with no symbol
  - A sequence of zero symbols
- Empty string is denoted by λ (pronounced "lambda").
- The length of λ:

$$|\lambda| = 0$$

  - In some books, empty string might be shown as: ε (epsilon)

Strings Operations

Concatenation

Definition

- Concatenation of two strings u and v is the string uv.

Example 5

- Let u = aaba and v = bb ; uv = ?
- uv = aababb

- The length of concatenation:

$$|uv| = |u| + |v|$$

- λ is the neutral element for concatenation:

$$\lambda w = w\lambda = w$$

Example 6

- λaabb = aabλb = aλabb = aλabbλ = aabb

Reverse

Definition

- Reverse of a string w is obtained by writing the symbols in reverse order.
- Reverse of w is denoted by w^R. (pronounced "w reverse")
  - If $w = a_1 a_2 \ldots a_{n-1} a_n$, then $w^R = a_n a_{n-1} \ldots a_2 a_1$

Example 7

- Let $w = aaba$; $w^R = ?$
- $w^R = abaa$

Homework

- Prove that $(uv)^R = v^R u^R$

Substring

Definition

- Substring of a string w is any string of consecutive symbols of w.

Example 8

String	Substring
aababb	aa
aababb	ab
aababb	bab
aababb	b
aababb	λ

Prefix and Suffix

Definition

- Let w be a sting. If we can write w = uv, then ...
  - u is called "prefix".
  - v is called "suffix".

Example 9

- Let w = aababb
- If we consider u = aa as a prefix for w, then the rest would be the suffix.
  - v = babb.

- Are these the only prefix and suffix?

Prefix and Suffix

Example 9 (cont'd)

- The complete list of all possible prefixes and suffixes of w are:

Prefix = u	Suffix = v
λ	aababb
a	ababb
aa	babb
aab	abb
aaba	bb
aabab	b
aababb	λ

- So, λ is prefix and suffix of every string (NOT at the same time). because: w = λ w = w λ

Exponent Operator

Definition

- For a string w and a natural number n, w^n is defined as concatenation of n w's.

$$w^n = \underbrace{w\ w\ w\ \dots\ w}_{n \text{ times}}$$

Example 10

- Let w = aaba ; $w^2 = ?$
- $w^2 = w\ w$ = aabaaaba
- $w^3 = w\ w\ w$ = aabaaabaaaba
- In general: $w\ w^n = w^n\ w = w^{n+1}$
- Where $n \in \mathbb{N}$ (natural numbers)

Special case

- $w^0 = ?$
- $w^0 = \lambda$
- How can you prove this?

Example 11

- $(aaba)^0 = \lambda$
- Note that $aaba^0 = aab$

Formal Languages

Introduction of Two New Operations on Sets

- Before introducing "formal languages", we need to introduce two new operations on sets.

- We did not mention them because we needed the concept of concatenation.

Star Operator on Alphabets

Definition

- Let Σ be an alphabet.
- Σ^* is the set of "all possible strings" obtained by concatenating "zero or more" symbols from Σ.

Example 12

- Let $\Sigma = \{a\}$; $\Sigma^* = ?$
- $\Sigma^* = \{a\}^* = \{\lambda, a, aa, aaa, aaaa, \ldots\}$

Star Operator on Alphabets

Example 13

- Let $\Sigma = \{a, b\}$; $\Sigma^* = ?$
- $\Sigma^* = \{a, b\}^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, ...\}$

- Note what strategy we used to enumerate all combinations?

- Let $\Sigma = \{a, b, c\}$; $\Sigma^* = ?$

Plus Operator on Alphabets

Definition

- Let Σ be an alphabet.
- Σ^+ is the set of "all possible strings" obtained by concatenating "one or more" symbols from Σ.

Example 14

- Let $\Sigma = \{a\}$; $\Sigma^+ = ?$
- $\Sigma^+ = \{a\}^+ = \{a, aa, aaa, aaaa, ...\}$

Plus Operator on Alphabets

Example 15

- Let $\Sigma = \{a, b\}$; $\Sigma^+ = ?$
- $\Sigma^+ = \{a, b\}^+ = \{a, b, aa, ab, ba, bb, aaa, aab, ...\}$

- Note that the only difference between Σ^+ and Σ^* is that Σ^+ does NOT contain λ.
- Hence:

$$\Sigma^+ = \Sigma^* - \{\lambda\}$$

$$\Sigma^* = \Sigma^+ \cup \{\lambda\}$$

- Also note that Σ is finite but both Σ^+ and Σ^* are infinite.

Formal Languages Definition

Definition

- Let Σ be an alphabet.
- Any subset of Σ^* is called a "formal language" over Σ.

- Σ^* is called the "universal formal language" over Σ.

Formal Languages Definition

Example 16

- Let $\Sigma = \{a, b\}$ be an alphabet.
- Then:
- $\Sigma^* = \{a, b\}^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, ...\}$

- The following sets are examples of formal languages.
- $L_1 = \{a, b, aa, aab\}$ because $L_1 \subseteq \Sigma^*$
- $L_2 = \{\lambda, ba, bb, bbb, aaa, aab\}$ because $L_2 \subseteq \Sigma^*$

Formal Languages

Example 16 (cont'd)

- How about the following sets? Are they languages? Why?
- $L_3 = \phi = \{\ \}$
- $L_4 = \{\lambda\}$

Two Special Languages

- Empty Language : $\{\ \}$ or ϕ
- Language with Empty String : $\{\lambda\}$

Formal Languages Notes

1. For simplicity, we use "language" to refer to the "formal language".
   – To refer natural languages, we specifically mention "natural" word.

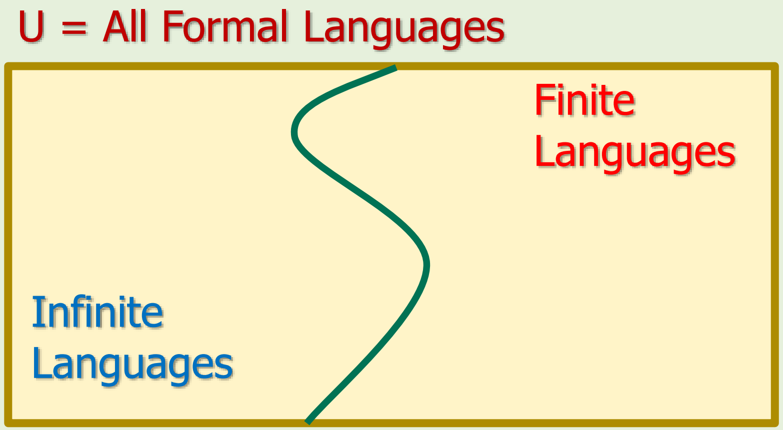
2. A language is a "set". So, it has all properties of sets.

3. $\{\lambda\}$ is a language while λ is a string.
   – $|\lambda| = 0$; This is the size of the string λ.
   – $|\phi| = |\{\ \}| = 0$; $|\{\lambda\}| = 1$; These are the sizes of languages.

Formal Languages Notes

4. In some books, "strings" are called "sentences" to analogize the formal languages with the natural languages.
   – In this course, we mostly use "strings"!

5. Like sets, we have both "finite" and "infinite" languages.
   – This is our first categorization of formal languages.



Formal Languages Exercises

Example 17

- Given the following languages by set-builder method over $\Sigma = \{a, b\}$.
- Represent them by using roster method (enumerate the strings):

- $L_1 = \{a^n b^n : n \geq 0\}$

This is our celebrity language!

- $L_2 = \{a^n b^{2n} : n \geq 0\}$
- $L_3 = \{a^{n+2} b^n : n \geq 0\}$
- $L_4 = \{a^n b^m : n \geq 0, m \geq 0\}$

References

1. Linz, Peter, "An Introduction to Formal Languages and Automata, 5th ed.," Jones & Bartlett Learning, LLC, Canada, 2012
2. Kenneth H. Rosen, "Discrete Mathematics and Its Applications, 7th ed.," McGraw Hill, New York, United States, 2012
3. Michael Sipser, "Introduction to the Theory of Computation, 3rd ed.," CENGAGE Learning, United States, 2013 ISBN-13: 978-1133187790