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

Fall 2019

Agenda of Day 04

- Waiting List Enrollment ...
- Summary of Lecture 03
- Lecture 04: Teaching ...
  - Formal Languages (Part 1)

Summary of Lecture 03: We learned ...

Cartesian Products

- We use Cartesian product when we need ordered collections.
- The Cartesian product of two sets A and B is ...
  - ... the set of all ordered pairs (a , b), where $a \in A$ and $b \in B$.

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

- It does NOT have commutative property.
- Special cases: $(A = B) \lor (A = \phi) \lor (B = \phi)$

- We extended the Cartesian product to n sets to produce n-tuple.

$$S_1 \times S_2 \times \ldots \times S_n = \{(x_1, x_2, \ldots , x_n) : x_1 \in S_1 , \ldots , x_n \in S_n\}$$

Any question?

Summary of Lecture 03: We learned ...

Functions

- We use Functions when we need a relation between two sets.
- A function f from D to R is ...
  - ... a rule that assigns to some elements of D (domain) a unique element of R (range).
  - Denoted by: $f : D \rightarrow R$
- 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 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 $V = \{v_1 , v_2 , \dots , v_n\}$
  - A finite set of edges $E = \{e_1 , e_2 , \dots , e_m\}$
- A walk is ...
  - ... a sequence of edges from v_i to v_n. $(v_i , v_j) , (v_j , v_k) , \dots , (v_m , v_n)$
- 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 vertices 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 vertices other than base is repeated.

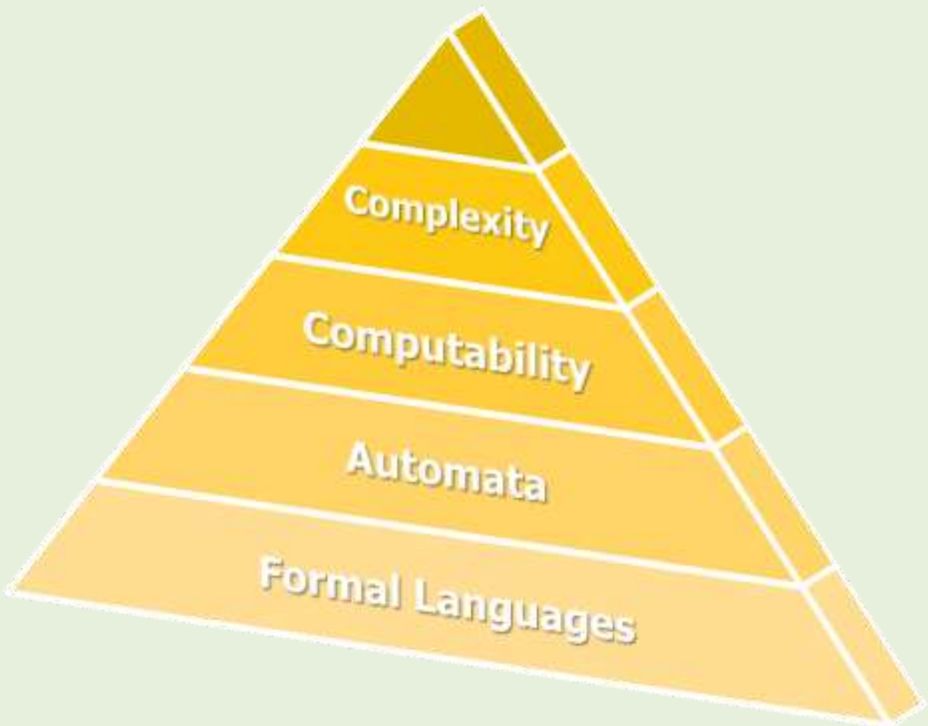
Any question?

The Big Picture of the Course

- The foundation of the computer science is called:

  "Theory of Computation".

- This theory is divided into four branches:



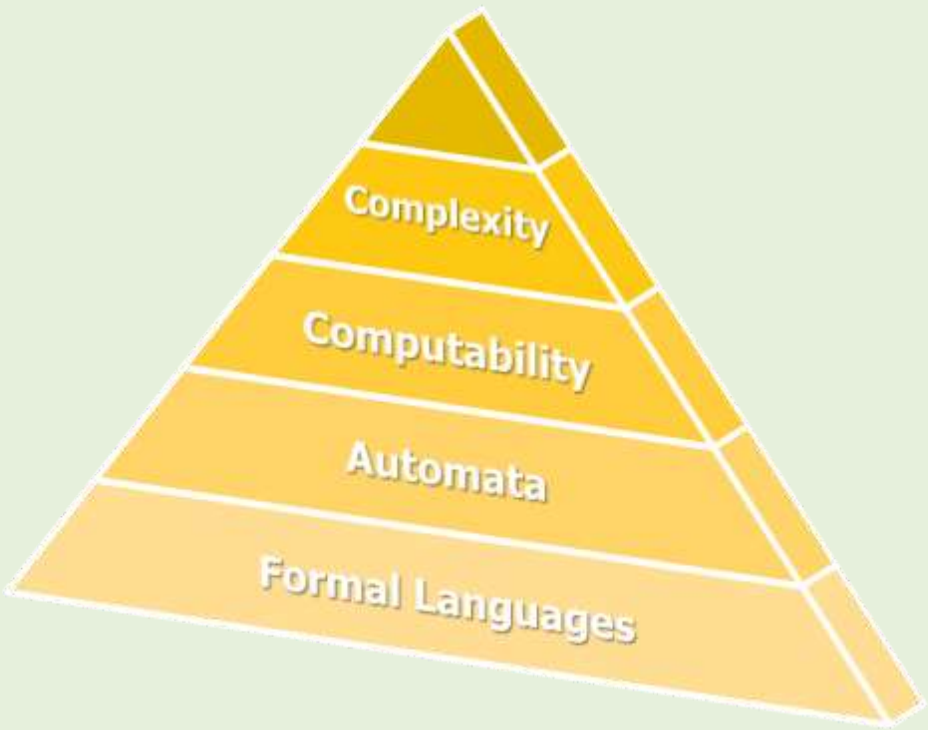
The Big Picture of the Course

- The first three from the bottom show:

  "What can be done with computers?"

- The forth one, complexity, shows:

  "What can be done in practice?"



Objective of This Lecture

- We start our journey with "Formal Languages"!
- But every language consists of "sentences" (aka "strings").
- And every string consists of "alphabets".

- So first, we'll introduce alphabets.
- Then, we'll define strings as mathematical objects.
- After that, we'll see some operations on strings.
- And finally, we'll introduce formal languages.

Alphabets & Strings

Alphabets

Definition

- An alphabet is a nonempty and finite set of "symbols".
- It is denoted by Σ.
  - Symbols are assumed to be indivisible.
- In this course, we usually use lowercase letters a, b, c, ... for alphabet symbols.
- In some cases, we might use digits like 0, and 1 or other symbols as well.

Alphabets Example

Example 1

- $\Sigma = \{a, b\}$ This is our celebrity alphabet!
- $\Sigma = \{0, 1\}$
- $\Sigma = \{\varepsilon, \alpha, \beta\}$

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

Strings

Definition

- A string is a finite sequence of symbols.
- So, we do NOT have a string of INFINITE symbols.

Example 2

Let $\Sigma = \{a, c, d, e, g, l, o, p, t\}$.

The following strings are valid strings over Σ:

cat , dog , apple

Strings Examples

Example 3

- Let Σ = {a, b}.
- Are the following strings valid strings over Σ?
- baba , aabb , bbbbbbbbbbbba , ...

Solution

- 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!

Strings Variable

- We usually use lowercase letters w, u, v, ... for "string variables".
  - If we need more variable, then x, y, z, or other symbols are used.

Example 4

- The following strings can be assigned to w as a variable:
- w = ab
- w = aabb
- w = aaabbb
- ...

Strings Size (aka Length)

Definition

- The size of a string w is the number of its symbols.
- It is denoted by |w|.

Example 5

|aaa| = 3

|babba| = 5

|aaba| = 4

- In general:

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

Empty String

Definition

- An empty string is a string with no symbol.
  - In other words: A sequence of zero symbols
- It is denoted by λ (pronounced lambda).

- What is the length of λ?

  $|\lambda| = 0$

Notes

1. Some authors might show empty string as ε (pronounced epsilon).
2. λ cannot be used as a symbol in alphabet.

Operations on Strings

Concatenation of Strings

Definition

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

Example 6

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 7

λaabb = aabλb = aλabb = aλabbλ = aabb

Reverse of Strings

Definition

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

Example 8

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

$w^R = abaa$

A Side Note: Palindrome

- The string w is called palindrome if w reads the same from left to right as from right to left.

Example 9

- radar, reviver, rotator

Some Funny Palindromes (Ignore spaces, apostrophes, commas)

- MADAM I'M ADAM
- STEP NOT ON PETS
- NO LEMONS, NO MELON
- DENNIS AND EDNA SINNED
- A MAN, A PLAN, A CANAL, PANAMA

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 10

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

Prefix and Suffix

Definition

- Let w be a string. If w = uv, then ...

  u is called "prefix".

  v is called "suffix".

Example 11

Let w = aababb

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

- Are these the only prefix and suffix?

Prefix and Suffix

Example 11 (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

Exponential Operator

Definition

- Let w be a string and n be a natural number.
- w^n is defined as the concatenation of n copies of w.

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

Example 12

Let $w = a$; $w^2 = ?$; $w^3 = ?$

$w^2 = w\ w = aa = a^2$

$w^3 = w\ w\ w = aaa = a^3$

- Concatenation in formal languages looks like multiplication in elementary algebra.

Exponential Operator

Example 13

Let $w = aaba$; $w^2 = ?$; $w^3 = ?$

$w^2 = w\,w = aabaaaba = a^2ba^3ba$

$w^3 = w\,w\,w = aabaaabaaaba = a^2ba^3ba^3ba$

- In general: $w\,w^n = w^n\,w = w^{n+1}$
  where $n \in \mathbb{N}$ (natural numbers)

Example 14

Let $w = a^m b^m$ where m is a constant.

$|w| = ?$

$|w| = |a^m b^m| = 2m$

Exponential Operator

Special case

- $w^0 = ?$
- $w^0 = \lambda$
- How can you prove this?
  - Hint: use $w\ w^n = w^n\ w = w^{n+1}$ and $w = \lambda\ w = w\ \lambda$

Example 15

$(aaba)^0 = \lambda$

- Note that $aaba^0 = aab$

Formal Languages

Introduction

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

- We did not mention them before 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 16

Let $\Sigma = \{a\}$; $\Sigma^* = ?$

$\Sigma^* = \{a\}^* = \{\lambda, a, aa, aaa, aaaa, ...\}$

- Note that Σ^* is an infinite set.

Star Operator on Alphabets

Example 17

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 18

Let $\Sigma = \{a\}$; $\Sigma^+ = ?$

$\Sigma^+ = \{a\}^+ = \{a, aa, aaa, aaaa, \ldots\}$

- Note that Σ^+ is an infinite set.
- Also, note that the only difference between Σ^+ and Σ^* is that Σ^+ does NOT contain λ.

Plus Operator on Alphabets

Example 19

Let $\Sigma = \{a, b\}$; $\Sigma^+ = ?$

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

- Since the only difference between Σ^+ and Σ^* is that Σ^+ does NOT contain λ, hence:

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

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

Formal Languages

Definition

- Let Σ be an alphabet.
- Any subset of Σ^* is called a "formal language" over Σ.
- Σ^* contains all possible strings that can be made by the symbols of Σ.
- That's why it's called the "universal formal language" over Σ.
  - Recall the definition of "universal set".

Formal Languages: Examples

Example 20

Let $\Sigma = \{a, b\}$ be an alphabet ; $\Sigma^* = ?$

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

- The following subsets of Σ^* are examples of formal languages over Σ:

- $L_1 = \{a, b, aa, aab\}$ because $L_1 \subseteq \Sigma^*$
- $L_2 = \{\lambda, ba, bb, bbb, aaa, aab\}$ because $L_2 \subseteq \Sigma^*$
- $L_3 = \{a^n : n \geq 0\}$ because $L_3 \subseteq \Sigma^*$
  - Can you enumerate L_3?

Formal Languages: Special Cases

Example 20 (cont'd)

How about the following sets? Are they formal languages? Why?

$L_4 = \phi = \{\ \}$

$L_5 = \{\lambda\}$

Yes they are because both are subsets of Σ^*.

Two Special Formal Languages

- Empty language : $\{\ \}$ or ϕ
- λ-Language : $\{\lambda\}$

Formal Languages: Notes

1. For simplicity, from now on, we use "language" to refer to the formal language.

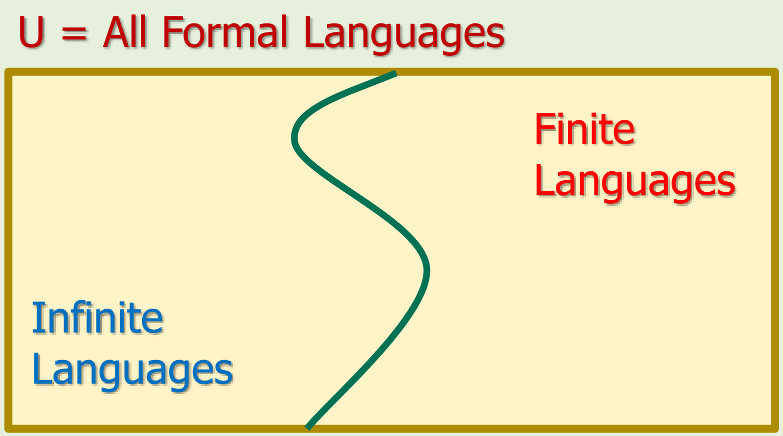
2. A language is a "set".
   So, it has all properties of sets.

3. $\{\lambda\}$ is a language while λ is a string.
   - Language is a set while string is a sequence of symbols.

4. Some authors prefer to call strings as "sentences" to analogize the formal languages with the natural languages.
   - In this course, we mostly use strings!

Languages Categorization: Finite and Infinite

- Like sets, we have both "finite" and "infinite" languages.



- We'll categorize languages from different angles.
- This is our first categorization that is from the size point of view.

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
4. Alfred S. Posamentier, "Math Wonders," Association for Supervision and Curriculum Development, Alexandria, Virginia, USA, 2003
   ISBN-13: 0-87120-775-3