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 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

- In many cases, we need ordered collections.
- We use Cartesian product to produce 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 $b \in B$.

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

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

- We extend 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

- 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 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 and Next Lecture

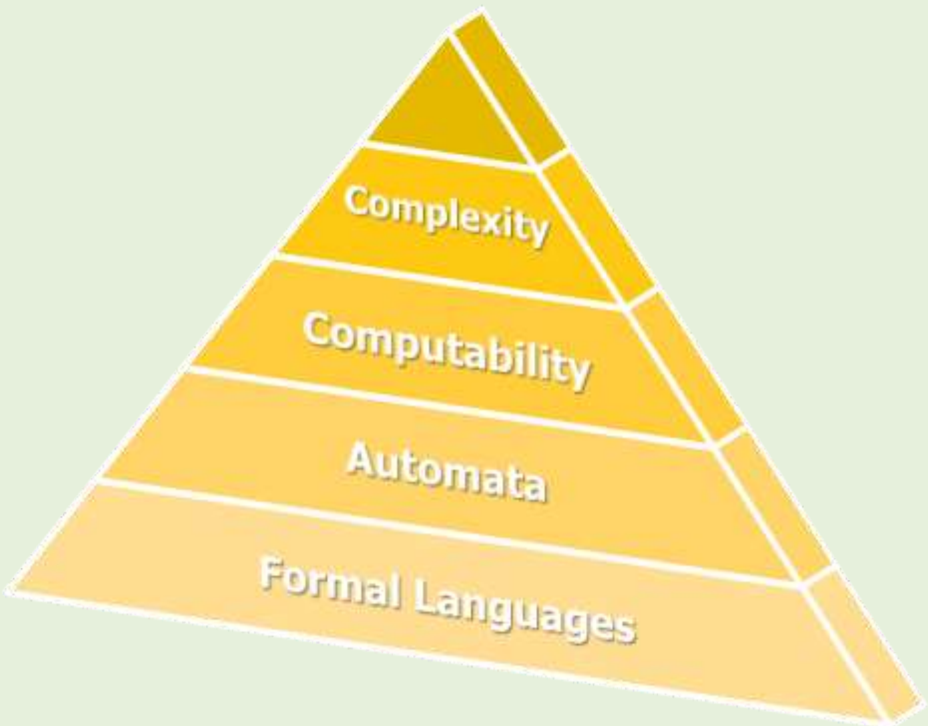
- Reviewing alphabets
- Reviewing strings
- Introducing formal languages
- Examining some surprising languages

The Big Picture of the Course

- The foundation of the computer science is called:

  "Theory of Computation".

- This theory is divided into four branches:



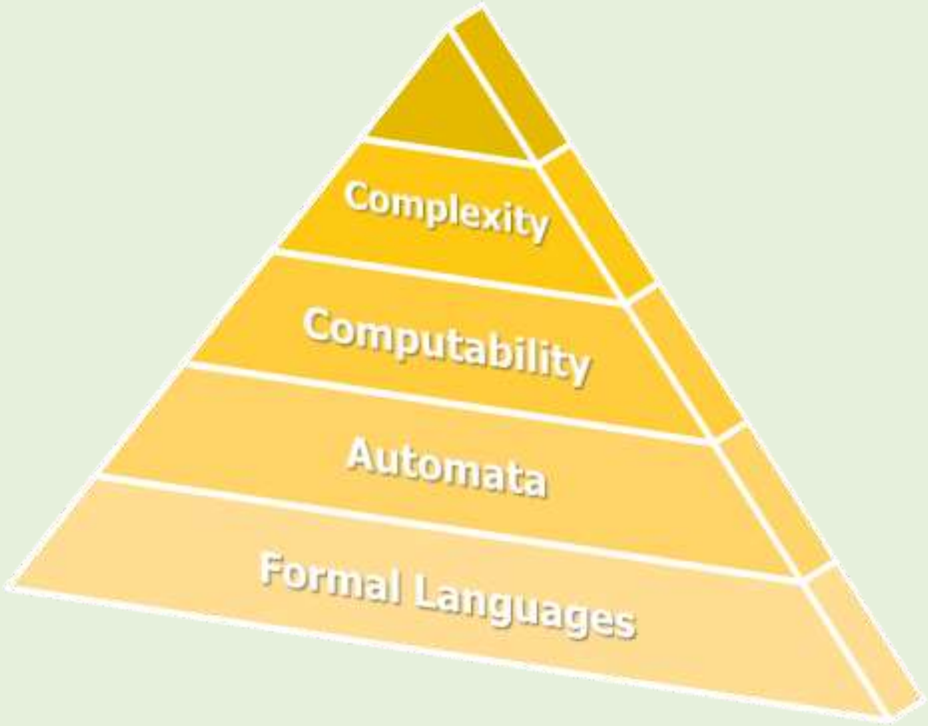
The Big Picture of the Course

- The first three from the bottom show:

  "What can be done with computers?"

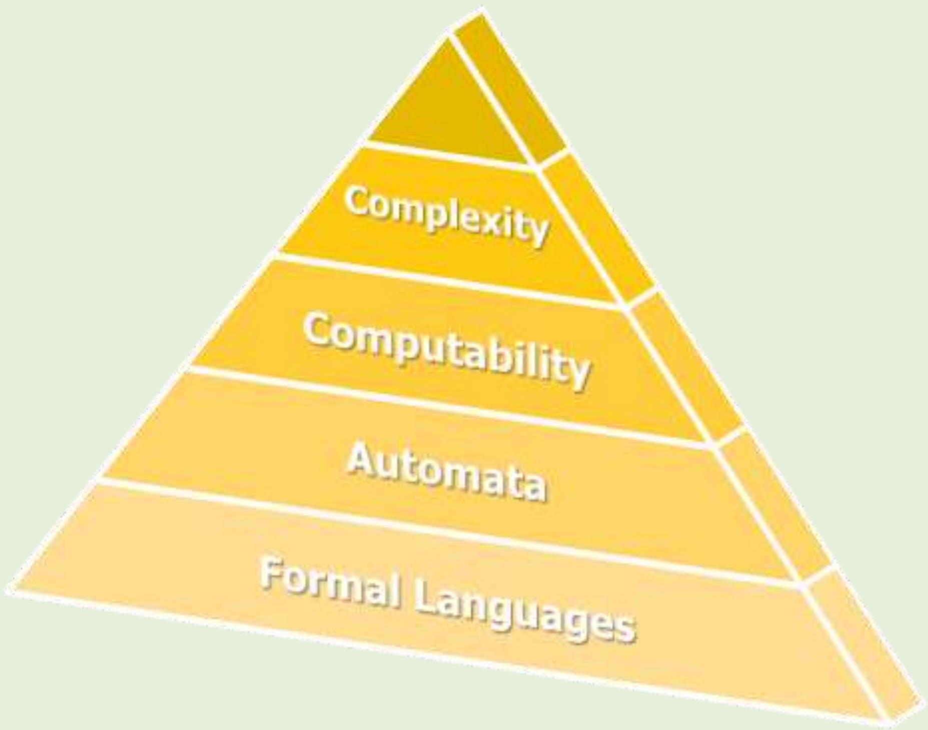
- The forth one, complexity, shows:

  "What can be done in practice?"



The Big Picture of the Course

- Let's start with "Formal Languages"!
- But first, we need to introduce "alphabets" and "strings".



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 use lowercase letters a, b, c, ... for alphabets' symbols.

- In some cases, we might use digits like 0, and 1 or other symbols as well.

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 = {Ԙ , Ѭ , پ , Γ }

Strings

Definition

- A string is a finite sequence of symbols from the alphabet.
- So, we do NOT have a string of infinite sequence of symbols.

Example 2

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

Are the following strings valid strings over Σ?

cat , dog , horse , house , apple

Strings Examples

Example 3

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

Strings Size (aka Length)

Definition

- The size of a string is the number of its symbols.
- 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

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

- What is the length of λ? $|\lambda| = ?$

  $|\lambda| = 0$

Notes

1. In some books, empty string might be shown as: ε (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 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

$\lambda aabb = aab\lambda b = a\lambda abb = a\lambda abb\lambda = aabb$

Reverse of Strings

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 $w = a_1 a_2 \dots a_{n-1} a_n$, then $w^R = a_n a_{n-1} \dots a_2 a_1$

Example 7

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.

Examples

- 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 8

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 9

Let w = aababb

If we consider u = aa as a prefix of 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

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.

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

Example 10

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 11

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 12

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 $w\,w^n = w^n\,w = w^{n+1}$

Example 13

$(aaba)^0 = \lambda$

- 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 14

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

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

Star Operator on Alphabets

Example 15

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 16

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

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

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

Plus Operator on Alphabets

Example 17

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:

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

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

- 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^* 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 Example

Example 18

Let $\Sigma = \{a, b\}$ be an alphabet.

Then:

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

- 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 $L_2 \subseteq \Sigma^*$

Formal Languages

Example 18 (cont'd)

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

$L_3 = \phi = \{\ \}$

$L_4 = \{\lambda\}$

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

Two Special Formal Languages

- Empty language : $\{\ \}$ or ϕ
- Language containing only empty string : $\{\lambda\}$

Formal Languages Notes

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

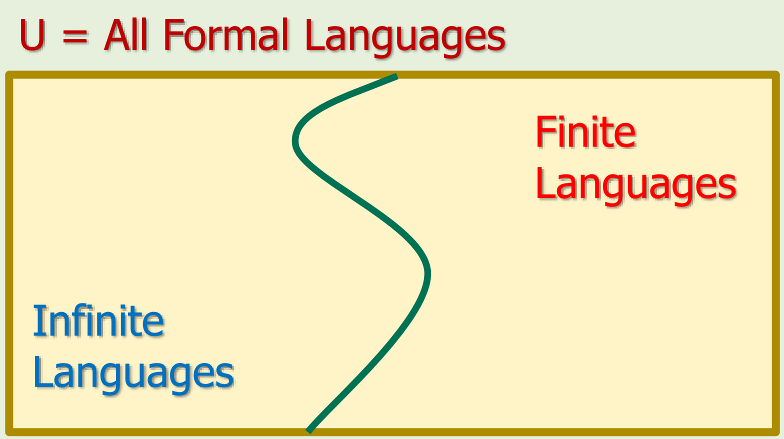
2. A language is a "set".
   So, it has all properties of sets.

3. $\{\lambda\}$ is a language while λ is a string.
   - $|\lambda| = 0$; This is the size of the string λ.
   - $|\{\lambda\}| = 1$

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 19

Given the following languages by set-builder over $\Sigma = \{a, b\}$.

Represent them by using roster method (enumerate the strings):

1. $L_1 = \{a^n b^n : n \geq 0\}$ This is our celebrity language!
2. $L_2 = \{a^n b^{2n} : n \geq 0\}$
3. $L_3 = \{a^{n+2} b^n : n \geq 0\}$
4. $L_4 = \{a^n b^m : n \geq 0, m \geq 0\}$

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