San José State University
Department of Computer Science

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

Formal Languages

(Part 2)

Lecture 05

Day 05/31

CS 154

Formal Languages and Computability

Spring 2019

Agenda of Day 05

- Rollcall Form and Your List Number
- Summary of Lecture 04
- Quiz 1
- Lecture 05: Teaching ...
  - Formal Languages (Part 2)

Summary of Lecture 04: We learned ...

Alphabets & Strings

- Alphabet is ...
  - ... a nonempty and finite set of symbols, denoted by Σ.
- String is ...
  - ... a finite sequence of symbols from the alphabet.
- Length of string w is ...
  - ... the number of symbols in the string, denoted by $|w|$.
- Empty string is ...
  - ... A string with no symbol, denoted by λ
  - $|\lambda| = 0$

Operations on Strings

- Concatenation of u and v is uv.
  - $\lambda w = w\lambda = w$ (neutral element)
- Reverse of w is denoted by w^R. (easy!)
- Substring (easy!)
- Prefix and Suffix
  - $w = uv$, u=prefix, v=suffix
  - λ is suffix and prefix of every string because: $w = \lambda\, w = w\, \lambda$
- Exponent operator
  - $w^n = w\, w\, w \ldots w$
  - $w\, w^n = w^n\, w = w^{n+1}$
  - $w^0 = \lambda$

Summary of Lecture 04: We learned ...

Formal Languages

- Star operator: Σ^*
  - The set of all possible strings obtained by concatenating zero or more symbols from Σ.
  - Universal set of all strings over Σ.
- Plus operator: Σ^+
  - The set of all possible strings obtained by concatenating one or more symbols from Σ.
  - $\Sigma^+ = \Sigma^* - \{\lambda\}$
  - $\Sigma^* = \Sigma^+ \cup \{\lambda\}$
- Formal language is ...
  - ... any subset of Σ^*
- Special cases:
  - { } and $\{\lambda\}$

- Formal languages are sets, so, they have all sets properties.
- Formal languages can be finite or infinite.
- This is the first categorization of languages:

U = All Formal Languages

Any question?

NAME	Alan M. Turing		
SUBJECT	CS 154	TEST NO.	1
DATE	02/07/2019	PERIOD	1 / 2 / 3

TEST RECORD	
PART 1	123
PART 2	
TOTAL	

Your list # goes here!

Quiz 1

Use Scantron

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. $L_4 = \{a^n b^m : n \geq 0, m \geq 0\}$

Operations on Languages

Regular Set Operations

- Since languages are sets,
  we can apply all regular set operations on them.
  - Note that the result of an operation is always a language.

Union

{a, aa, ab} ∪ {a, ab, bbb, bba, b} = {a, aa, ab, bbb, bba, b}

Intersection

{a, aa, ab} ∩ {a, ab, bbb, bba, b} = {a, ab}

Minus

{a, aa, ab} - {a, ab, bbb, bba, b} = {aa}

Complement of Languages

Definition

- Let L be a language over Σ.
- Complement of L, denoted by $\overline{L}$, is defined as:

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

Example 20

Let $L = \{\lambda, b, aa, aab\}$ over $\Sigma = \{a, b\}$; $\overline{L} = ?$

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

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

$$\overline{L} = \{\cancel{\lambda}, a, \cancel{b}, \cancel{aa}, ab, ba, bb, aaa, \cancel{aab}, \ldots\}$$

$$= \{a, ab, ba, bb, aaa, \ldots\}$$

Homework

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

1. Set of all strings that contains at least one a
2. Set of all strings that contains more than one a
3. Set of all strings that contains exactly one a

Reverse of Languages

Definition

- Let L be a language over a given alphabet Σ.
- Reverse of L, denoted by L^R, is defined as:

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

Example 21

Let $L = \{b, ab, aab, abab\}$; $L^R = ?$

$L^R = \{b, ba, baa, baba\}$

Example 22

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

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

Concatenation of Languages

Definition

- Let L_1 and L_2 be two languages over Σ_1 and Σ_2.
- The concatenation of L_1 and L_2, denoted by L_1L_2, is a language defined as:

$$L = L_1L_2 = \{xy : x \in L_1, y \in L_2\} \text{ over } \Sigma = \Sigma_1 \cup \Sigma_2$$

Example 23

Let $L_1 = \{a, ab\}$ over $\Sigma_1 = \{a, b\}$ and
$L_2 = \{c, ca, caa\}$ over $\Sigma_2 = \{a, c\}$; $L_1L_2 = ?$

$L_1L_2 = \{a, ab\} \{c, ca, caa\}$

$= \{ac, aca, acaa, abc, abca, abcaa\}$

Over $\Sigma = \Sigma_1 \cup \Sigma_2 = \{a, b\} \cup \{a, c\} = \{a, b, c\}$

Concatenation Notes

1. The concatenation of two languages looks like Cartesian product of two sets.
   - Instead of ordered-pair, we concatenate two strings.
2. $\phi \, L = L \, \phi = \phi$ (prove it!)
   - ϕ has the same role as 0 (zero) for multiplication.

3. $\{\lambda\} \, L = L \, \{\lambda\} = L$
   - $\{\lambda\}$ is the neutral language for concatenation operation.
   - $\{\lambda\}$ has the same role as number 1 (one) for multiplication.
4. Properties of concatenation:

$$L \, (L_1 \cup L_2) = L \, L_1 \cup L \, L_2 \; ; \; (L_1 \cup L_2) \, L = L_1 \, L \cup L_2 \, L$$

$$L \, (L_1 \cap L_2) = L \, L_1 \cap L \, L_2 \; ; \; (L_1 \cap L_2) \, L = L_1 \, L \cap L_2 \, L$$

Exponential Operator

Definition

- Let L be a language and n be a natural number.
- L^n is defined as concatenation of n copies of L's.

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

Example 24

Let $L = \{a, ab\}$; $L^2 = ?$; $L^3 = ?$

$L^2 = \{a, ab\}\ \{a, ab\}$

$= \{aa, aab, aba, abab\}$

$L^3 = L\ L^2 = \{a, ab\}\ \{aa, aab, aba, abab\}$

$= \{aaa, aaab, aaba, aabab, abaa, abaab, ababa, ababab\}$

Exponential Operator

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

Example 25

Let $L = \{a^n b^n : n \geq 0\}$; $L^2 = ?$

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

- Note that n and m are independent.
- For example abaabb (n=1, m=2) belongs to L^2.
- How about $L^3 = ?$

Homework

Special cases

- $L^0 = ?$
- $L^0 = \{\lambda\}$ (prove it!)
- $\phi^0 = ?$

Homework

- Enumerate at least 5 elements of the following languages:

1. $L = \{w \in \{a, b\}^+\}$
2. $L = \{w \in \{a, b\}^+ : |w| = 2k, K \geq 0\}$
3. $L = \{w \in \{a, b\}^+ : |w| = 2k+1, K \geq 0\}$
4. $L = \{1^{2k} : k \geq 1\}$ over $\Sigma = \{1\}$
5. $L = \{w \in \{a, b\}^+ : n_a(w) = n_b(w)\}$ //number of a's = number of b's
6. $L = \{a^n b^n c^n : n \geq 1\}$
7. $L = \{a^n b^m c^{nm} : n, m \geq 1\}$
8. $L = \{w\#w : w \in \{a, b\}^+\}$
9. $L = \{w \in \{a, b\}^+ : |w| = 2k+1, K \geq 0,$ w contains at least one a$\}$
10. $L = \{ww : w \in \{a, b\}^+\}$

Surprising Languages

Surprising Languages

- In this section, we look at some familiar sets of objects from different angle.
- Based on formal languages definition, they seems to be formal languages!
- We'll introduce a new set of numbers called "Unary Numbers".

Surprising Languages Examples

Example 26 : Natural Numbers

Consider the set of natural numbers:

$\mathbb{N} = \{0, 1, 2, 3, 4, \ldots, 123, \ldots, 456, \ldots, 5908764, \ldots\}$

Can we consider $\mathbb{N}$ as a formal language over $\Sigma = \{0, 1, \ldots, 9\}$?

Yes, numbers are just sequence of digits that can be considered as symbols!

Surprising Languages Examples

Example 27 : Binary$^+$ Numbers

Consider the set of positive binary numbers:

B = {0, 1, ..., 1010, ..., 10000001, ..., 111100001, ...}

Can we consider B as a formal language
over Σ = {0, 1}?

Yes, for the same reason we saw for natural numbers!

Surprising Languages Examples

- How about the following sets?

Example 28: Prime Numbers

$\Sigma = \{0, 1, 2, ..., 9\}$

$L = \{2, 3, 5, 7, 11, 13, 17, ...\}$

Example 29: Even and Odd Numbers

$\Sigma = \{0, 1, 2, ..., 9\}$

$L_1 = \{0, 2, 4, 6, 8, ...\}$

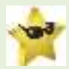
$L_2 = \{1, 3, 5, 7, 9, ...\}$

- Yes, for the same reasons!

Introducing Unary Numbers

Definition

- Given $\Sigma = \{1\}$.
- We define the following set as "unary numbers":

 $A = \{1, 11, 111, 1111, 11111, ...\}$

This is our celebrity numbers!

- Equivalent natural numbers: $\{1, 2, 3, 4, 5, ... \}$

- How can we show the unary numbers by set builder?

Surprising Languages Examples

Example 30: Addition of Unary Numbers

$L = \{1^n+1^m=1^{n+m} : n \geq 1, m \geq 1\}$

Over $\Sigma = \{1, +, =\}$

In computer science,
all data are strings!

Membership: L contains strings such as:

1+11=111

11+111=11111

...

Not Membership: L doesn't contain strings such as:

1+11=1

11+111=11

Homework

Square of Unary Numbers

$L = \{1^n\#=1^k : k = n^2, n \geq 1\}$

Over $\Sigma = \{1, \#, =\}$

Membership: L contains strings such as:

??

Not Membership: L doesn't contain strings such as:

??

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