

بسم الله الرحمن الرحيم

## فصل اول

# مقدمه‌ای بر نظریه‌ی محاسبات (۲)

An Introduction to the Theory of Computation (2)

کاظم فولادی

kazim@fouladi.ir

دانشکده‌ی مهندسی برق و کامپیوتر

دانشگاه تهران



# Languages

# Languages

A language is a set of strings

String: A sequence of letters

Examples: "cat", "dog", "house", ...

Defined over an alphabet:

$$\Sigma = \{a, b, c, \dots, z\}$$

# Alphabets and Strings

We will use small alphabets:  $\Sigma = \{a, b\}$

## Strings

*a*

*ab*

$u = ab$

*abba*

$v = bbbaaa$

*baba*

$w = abba$

*aaabbbbaabab*

# String Operations

$$w = a_1 a_2 \cdots a_n$$

*abba*

$$v = b_1 b_2 \cdots b_m$$

*bbbbaaa*

## Concatenation

$$wv = a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m$$

*abbabbbaaa*

$$w = a_1 a_2 \cdots a_n$$

*ababaaabbb*

Reverse

$$w^R = a_n \cdots a_2 a_1$$

*bbbaaababa*

# String Length

$$w = a_1 a_2 \cdots a_n$$

Length:  $|w| = n$

Examples:  $|abba| = 4$

$$|aa| = 2$$

$$|a| = 1$$

# Recursive Definition of Length

For any letter:  $|a| = 1$

For any string  $wa$ :  $|wa| = |w| + 1$

Example:  $|abba| = |abb| + 1$   
 $= |ab| + 1 + 1$   
 $= |a| + 1 + 1 + 1$   
 $= 1 + 1 + 1 + 1$   
 $= 4$



# Length of Concatenation

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

Example:  $u = aab, |u| = 3$

$v = abaab, |v| = 5$

$$|uv| = |aababaab| = 8$$

$$|uv| = |u| + |v| = 3 + 5 = 8$$

# Proof of Concatenation Length

**Claim:**  $|uv| = |u| + |v|$

**Proof:** By induction on the length  $|v|$

**Induction basis:**  $|v| = 1$

From definition of length:

$$|uv| = |u| + 1 = |u| + |v|$$

Inductive hypothesis:  $|uv| = |u| + |v|$

for  $|v| = 1, 2, \dots, n$

Inductive step: we will prove  $|uv| = |u| + |v|$

for  $|v| = n + 1$

## Inductive Step

Write  $v = wa$ , where  $|w| = n$ ,  $|a| = 1$

From definition of length:  $|uv| = |uwa| = |uw| + 1$

$$|wa| = |w| + 1$$

From inductive hypothesis:  $|uw| = |u| + |w|$

Thus:  $|uv| = |u| + |w| + 1 = |u| + |wa| = |u| + |v|$

# Empty String

A string with no letters:  $\lambda$

Observations:  $|\lambda| = 0$

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

$$\lambda abba = abba \lambda = abba$$

# Substring

Substring of string:

a subsequence of consecutive characters

String

abbab

abba

abba

bbab

Substring

ab

abba

b

bbab

# Prefix and Suffix

*abbab*

Prefixes

Suffixes

$\lambda$

*abbab*

*a*

*bbab*

*ab*

*bab*

*abb*

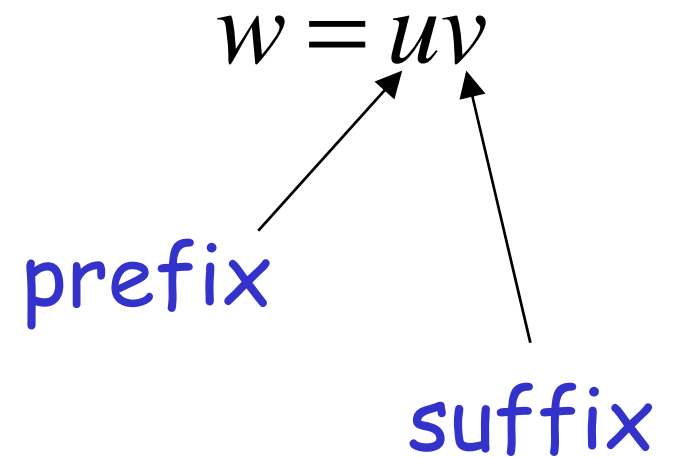
*ab*

*abba*

*b*

*abbab*

$\lambda$



## Another Operation

$$w^n = \underbrace{ww \cdots w}_n$$

Example:  $(abba)^2 = abbaabba$

Definition:  $w^0 = \lambda$

$$(abba)^0 = \lambda$$



# The \* Operation

$\Sigma^*$ : the set of all possible strings from alphabet  $\Sigma$

$$\Sigma = \{a, b\}$$

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

# The + Operation

$\Sigma^+$  : the set of all possible strings from alphabet  $\Sigma$  except  $\lambda$

$$\Sigma = \{a, b\}$$

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

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

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

# Language

A language is any subset of  $\Sigma^*$

Example:  $\Sigma = \{a, b\}$

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

Languages:  $\{\lambda\}$

$$\{a, aa, aab\}$$

$$\{\lambda, abba, baba, aa, ab, aaaaaa\}$$

## Another Example

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

$\lambda$   
 $ab$   
 $aabb$   
 $aaaaabbbbb$

}  $\in L$        $abb \notin L$

# Operations on Languages

## The usual set operations

$$\{a, ab, aaaa\} \cup \{bb, ab\} = \{a, ab, bb, aaaa\}$$

$$\{a, ab, aaaa\} \cap \{bb, ab\} = \{ab\}$$

$$\{a, ab, aaaa\} - \{bb, ab\} = \{a, aaaa\}$$

Complement:  $\bar{L} = \Sigma^* - L$

$$\overline{\{a, ba\}} = \{\lambda, b, aa, ab, bb, aaa, \dots\}$$

# Reverse

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

Examples:  $\{ab, aab, baba\}^R = \{ba, baa, abab\}$

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

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

# Concatenation

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

Example:  $\{a, ab, ba\}\{b, aa\}$

$= \{ab, aaa, abb, abaa, bab, baaa\}$

## Another Operation

Definition:  $L^n = \underbrace{LL \cdots L}_n$

$$\{a, b\}^3 = \{a, b\} \{a, b\} \{a, b\} = \\ \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$

Special case:  $L^0 = \{\lambda\}$

$$\{a, bba, aaa\}^0 = \{\lambda\}$$



# More Examples

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

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

$$aabbbaaabb \in L^2$$

# Star-Closure (Kleene \*)

Definition:  $L^* = L^0 \cup L^1 \cup L^2 \dots$

Example:

$$\{a, bb\}^* = \left\{ \begin{array}{l} \lambda, \\ a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \dots \end{array} \right\}$$

# Positive Closure

Definition:  $L^+ = L^1 \cup L^2 \cup \dots$   
 $= L^* - \{\lambda\}$

$$\{a, bb\}^+ = \left\{ \begin{array}{l} a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \dots \end{array} \right\}$$

# Grammars

# Grammars

Grammars express languages

Example: the English language

$$\langle sentence \rangle \rightarrow \langle noun\_phrase \rangle \langle predicate \rangle$$
$$\langle noun\_phrase \rangle \rightarrow \langle article \rangle \langle noun \rangle$$
$$\langle predicate \rangle \rightarrow \langle verb \rangle$$

$\langle \textit{article} \rangle \rightarrow a$

$\langle \textit{article} \rangle \rightarrow \textit{the}$

$\langle \textit{noun} \rangle \rightarrow \textit{boy}$

$\langle \textit{noun} \rangle \rightarrow \textit{dog}$

$\langle \textit{verb} \rangle \rightarrow \textit{runs}$

$\langle \textit{verb} \rangle \rightarrow \textit{walks}$

A derivation of "the boy walks":

$\langle sentence \rangle \Rightarrow \langle noun\_phrase \rangle \langle predicate \rangle$   
 $\Rightarrow \langle noun\_phrase \rangle \langle verb \rangle$   
 $\Rightarrow \langle article \rangle \langle noun \rangle \langle verb \rangle$   
 $\Rightarrow the \langle noun \rangle \langle verb \rangle$   
 $\Rightarrow the \ boy \langle verb \rangle$   
 $\Rightarrow the \ boy \ walks$

A derivation of "a dog runs":

$\langle sentence \rangle \Rightarrow \langle noun\_phrase \rangle \langle predicate \rangle$   
 $\Rightarrow \langle noun\_phrase \rangle \langle verb \rangle$   
 $\Rightarrow \langle article \rangle \langle noun \rangle \langle verb \rangle$   
 $\Rightarrow a \langle noun \rangle \langle verb \rangle$   
 $\Rightarrow a \text{ dog } \langle verb \rangle$   
 $\Rightarrow a \text{ dog runs}$



Language of the grammar:

$$L = \{ \text{"a boy runs"}, \\ \text{"a boy walks"}, \\ \text{"the boy runs"}, \\ \text{"the boy walks"}, \\ \text{"a dog runs"}, \\ \text{"a dog walks"}, \\ \text{"the dog runs"}, \\ \text{"the dog walks"} \}$$

# Notation

$\langle noun \rangle \rightarrow boy$

$\langle noun \rangle \rightarrow dog$

Variable  
or

Non-terminal

Production  
rule

Terminal

## Another Example

Grammar:  $S \rightarrow aSb$

$S \rightarrow \lambda$

Derivation of sentence  $ab$ :

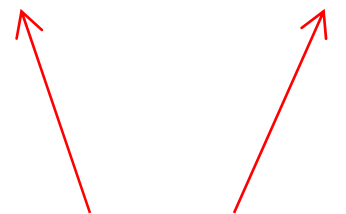
$$\begin{array}{ccc} & S \Rightarrow aSb \Rightarrow ab & \\ \nearrow & & \nwarrow \\ S \rightarrow aSb & & S \rightarrow \lambda \end{array}$$

Grammar:  $S \rightarrow aSb$

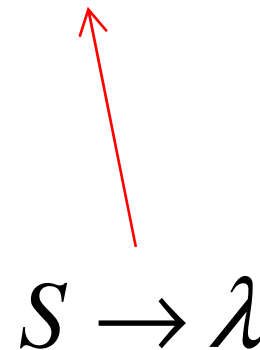
$S \rightarrow \lambda$

Derivation of sentence  $aabb$  :

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



$S \rightarrow aSb$



$S \rightarrow \lambda$

Other derivations:

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbbb$$

$$\begin{aligned} S &\Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \\ &\Rightarrow aaaaSbbbb \Rightarrow aaabbbbb \end{aligned}$$

Language of the grammar

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

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

# More Notation

**Grammar**       $G = (V, T, S, P)$

$V$  :    Set of variables

$T$  :    Set of terminal symbols

$S$  :    Start variable

$P$  :    Set of Production rules

# Example

Grammar  $G$  :  $S \rightarrow aSb$   
 $S \rightarrow \lambda$

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

$$V = \{S\}$$

$$T = \{a, b\}$$

$$P = \{S \rightarrow aSb, S \rightarrow \lambda\}$$



# More Notation

## Sentential Form:

A sentence that contains  
variables and terminals

Example:

$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbbb$

Sentential Forms

sentence

We write:  $S \xRightarrow{*} aaabbb$

Instead of:

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaasbbb \Rightarrow aaabbbb$$

In general we write:  $w_1 \overset{*}{\Rightarrow} w_n$

If:  $w_1 \Rightarrow w_2 \Rightarrow w_3 \Rightarrow \cdots \Rightarrow w_n$

By default:

$$w \stackrel{*}{\Rightarrow} w$$

# Example

## Grammar

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

## Derivations

$$\begin{array}{c} * \\ S \Rightarrow \lambda \end{array}$$

$$\begin{array}{c} * \\ S \Rightarrow ab \end{array}$$

$$\begin{array}{c} * \\ S \Rightarrow aabb \end{array}$$

$$\begin{array}{c} * \\ S \Rightarrow aaabbb \end{array}$$

# Example

## Grammar

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

## Derivations

$$S \xRightarrow{*} aaSbb$$

$$aaSbb \xRightarrow{*} aaaaaaSbbbbbb$$

# Another Grammar Example

Grammar  $G$  :  $S \rightarrow Ab$   
 $A \rightarrow aAb$   
 $A \rightarrow \lambda$

Derivations:

$$S \Rightarrow Ab \Rightarrow b$$

$$S \Rightarrow Ab \Rightarrow aAbb \Rightarrow abb$$

$$S \Rightarrow aAbb \Rightarrow aaAbbb \Rightarrow aabbbb$$

## More Derivations

$$S \Rightarrow Ab \Rightarrow aAbb \Rightarrow aaAbbb \Rightarrow aaaAbbbb \\ \Rightarrow aaaaAbbbbbb \Rightarrow aaaaabbbbbb$$

$$\begin{array}{c} * \\ S \Rightarrow aaaaabbbbbb \end{array}$$

$$\begin{array}{c} * \\ S \Rightarrow aaaaaabbbbbbbb \end{array}$$

$$\begin{array}{c} * \\ S \Rightarrow a^n b^n b \end{array}$$



# Language of a Grammar

For a grammar  $G$   
with start variable  $S$  :

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

String of terminals

## Example

For grammar  $G$  :

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

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

Since:  $S \xRightarrow{*} a^n b^n b$

# A Convenient Notation

$$\begin{array}{l} A \rightarrow aAb \\ A \rightarrow \lambda \end{array} \quad \longrightarrow \quad A \rightarrow aAb \mid \lambda$$

$$\begin{array}{l} \langle article \rangle \rightarrow a \\ \langle article \rangle \rightarrow the \end{array} \quad \longrightarrow \quad \langle article \rangle \rightarrow a \mid the$$