Module 1

Introduction to Languages and Grammars

Recall on Proof techniques in Mathematics - Overview of a Computational Models - Languages and Grammars - Alphabets - Strings - Operations on Languages, Overview on Automata

Why do we study Theory of Computation ?

- Understanding the Limits of Computation: To know which problem can be algorithmically solved and which cannot.
- Algorithmic Efficiency: Helps in analyzing the efficiency of algorithms.
- Formal Models of Computation: Turing machines, Pushdown Automata and Finite Automata
- Language Recognition and Automata Theory: Understanding how machines process languages.
- Compiler Design and Language Processing

Recall on Proof techniques in Mathematics

- 1. Direct Proof (Proof by Construction)
  - In a constructive proof, $P \Rightarrow Q$ is demonstrated directly.
  - Simplest and easiest method of proof
  - 2 steps:
    - 1. Assume that P is true.
    - 2. Use P to show that Q must be true.

Recall on Proof techniques in Mathematics

- If a and b are consecutive integers, then the sum $a + b$ is odd.
  - Part 1: Assume that a and b are consecutive integers.
  - Part 2: Because a and b are consecutive, we know that $b = a + 1$.
  - Hence, $a + b = a+a+1 = 2a+1$.
  - $\Rightarrow$ the sum $a+b$ is odd.

Recall on Proof techniques in Mathematics

- 2. Proof by Contradiction
  - Based on the fact that any proposition must be either true or false, but not both
  - Use this to demonstrate $P \Rightarrow Q$ by assuming both P and $\neg Q$ are simultaneously true and deriving a contradiction.
  - When we derive this contradiction it means that one of our assumptions was incorrect.
  - We have either assumed or already proved P to be true. Hence, finding a contradiction implies that $\neg Q$ must be false.

Recall on Proof techniques in Mathematics

- 2. Proof by Contradiction (Contd.)
  - The method of proof by contradiction.

    1. Assume that P is true.

    2. Assume that ¬Q is true.

    3. Use P and ¬Q to demonstrate a contradiction.

Recall on Proof techniques in Mathematics

- 2. Proof by Contradiction (contd.)
  - Theorem: If a and b are consecutive integers, then the sum $a + b$ is odd.
  - Proof
    - 1. Assume that a and b are consecutive integers.
    - 2. Assume that the sum $a+b$ is not odd.
    - 3. Because $a+b$ is not odd, there exists no number k such that $a + b = 2k + 1$. However, the integers a and b are consecutive, so $a+b = a+a+1 = 2a+1$.

      i.e., $a + b \mathrel{!=} 2k + 1$ for any integer k and also $a + b = 2a + 1$. This is a contradiction. Hence, second assumption is wrong. The sum of two consecutive integers is odd.

Recall on Proof techniques in Mathematics

- 3. Proof by Induction
- Powerful method that uses recursion to demonstrate an infinite number of facts in a finite amount of space.
- First create a propositional form whose truth is determined by an integer function. If we can show that the propositional form is true for some integer value then we may argue from that basis that the propositional form must be true for all integers.

1. Show that a propositional form P(x) is true for a basis case.
2. Assume that P(n) is true for some n, and show that this implies that P(n + 1) is true.
3. Then, by the principle of induction, the propositional form P(x) is true for all n greater or equal to the basis case.

Recall on Proof techniques in Mathematics

❖ 3. Proof by Induction (contd.)

Theorem: If a and b are consecutive integers, then the sum $a + b$ is odd.

Proof: Define the propositional form $F(x)$ to be true when the sum of x and its successor is odd.

1. (Basis) Consider the proposition $F(1)$.

The sum $1 + 2 = 3$ is odd because there exists an integer k such that $2k + 1 = 3$. i.e., $2(1)+1=3$. Thus, $F(x)$ is true when $x=1$.

Recall on Proof techniques in Mathematics

- 3. Proof by Induction (contd.)
  - 2. Assume that F(x) is true for some x. i.e., for some x we have that x+(x+1) is odd. (i.e., 2x+1)

    Now, to show that F(x+1) is true: We add one to both x and x+1 which gives the sum (x+1)+(x+2).

    First, we claim that the sum (x+1)+(x+2) = F(x+1). (i.e., (2x+1)+2) Second, we claim that adding 2 to any integer does not change the integer's evenness or oddness. With these two observations, F(x) is odd implies F (x + 1) is odd.
  - 3. By the principle of mathematical induction we thus claim that F(x) is odd for all integers x. So, the sum of two consecutive numbers is odd.

Recall on Proof techniques in Mathematics

- 4. Proof by Contrapositive
  - $P \Rightarrow Q$ is equivalent to $\neg Q \Rightarrow \neg P$
  - $\neg Q \Rightarrow \neg P$: contrapositive of $P \Rightarrow Q$
  - By saying that the two propositions are equivalent, we mean that if one can prove $P \Rightarrow Q$ then they have also proved $\neg Q \Rightarrow \neg P$, and vice versa.

  - Proof by contraposition can be an effective approach when a direct proof is tricky, or it can be a different way to think about the problem.

Recall on Proof techniques in Mathematics

- Theorem: If a and b are consecutive integers, then the sum a + b is odd.
- Contrapositive: If the sum a + b is not odd, then a and b are not consecutive integers.
- Proof:
  - Assume that the sum of the integers a and b is not odd.
  - Then, there exists no integer k such that a + b = 2k + 1.
  - Thus, a + b != k + (k + 1) for all integers k.
  - Because k + 1 is the successor of k, this implies that a and b cannot be consecutive integers.

Recall on Proof techniques in Mathematics

- Theorem: If $7x + 9$ is even, proof x is odd.

Recall on Proof techniques in Mathematics

- Theorem: If A is true, proof B $\wedge$ C
  - $\sim(B \wedge C) \rightarrow \sim A$
  - $\sim B \vee \sim C \rightarrow \sim A$
  - Only one among ~B and ~C is enough.

Recall on Proof techniques in Mathematics

- 5. Deductive Proofs
  - From the given statement(s) to a conclusion statement (what we want to prove)
  - Logical progression by direct implications
  - Example for parsing a statement:
    - If $y \geq 4$, then $2^y \geq y^2$.

Recall on Proof techniques in Mathematics

- 5. Deductive Proofs
  - If $y \geq 4$, then $2^y \geq y^2$.
    - For $y = 1$, $2 \geq 1$ … Yes
    - For $y = 3$, $8 \geq 9$ … No
    - For $y = 4$, $16 \geq 16$ … Yes
    - For $y = 5$, $32 \geq 25$ … Yes
    - Let's take the Ratio,
    - $2^{y+1}/2^y$ and $(y+1)^2/y^2$
    - RHS can never be greater than 1.56.
    - Since $2 > 1.56$, the $2^y \geq y^2$ is true for $y \geq 4$.

Set theory

Symbol	Symbol Name	Meaning	Example
{ }	set	a collection of elements	A = {1, 7, 9, 13, 15, 23}, B = {7, 13, 15, 21}
A ∪ B	union	Elements that belong to set A or set B	A ∪ B = {1, 7, 9, 13, 15, 21, 23}
A ∩ B	intersection	Elements that belong to both the sets, A and B	A ∩ B = {7, 13, 15 }
A ⊆ B	subset	subset has few or all elements equal to the set	{7, 15} ⊆ {7, 13, 15, 21}
A ⊄ B	not subset	left set is not a subset of right set	{1, 23} ⊄ B

Set theory

Symbol	Symbol Name	Meaning	Example
A ⊂ B	proper subset / strict subset	subset has fewer elements than the set	{7, 13, 15} ⊂ {1, 7, 9, 13, 15, 23}
A ⊃ B	proper superset / strict superset	set A has more elements than set B	{1, 7, 9, 13, 15, 23} ⊃ {7, 13, 15, }
A ⊇ B	superset	set A has more elements or equal to the set B	{1, 7, 9, 13, 15, 23} ⊇ {7, 13, 15, 23}
Ø	empty set	Ø = { }	C = {}
P (C)	power set	all subsets of C	C = {4,7}, P(C) = {{}, {4}, {7}, {4,7}} Given by 2^s, s is number of elements in set C
A ⊅ B	not superset	set X is not a superset of set Y	{1, 2, 5} ⊅{1, 6}
A = B	equality	both sets have the same members	{7, 13,15} = {7, 13, 15}

Set theory

Symbol	Symbol Name	Meaning	Example
$a \in B$	element of	set membership	$B = \{7, 13, 15, 21\}$, $13 \in B$
(a, b)	ordered pair	collection of 2 elements	(1, 2)
$x \notin A$	not element of	no set membership	$A = \{1, 7, 8, 13, 15, 23\}$, $5 \notin A$
$\|B\|$	cardinality	the number of elements of set B	$B = \{7, 13, 15, 21\}$, $\|B\| = 4$

Alphabet and String

- Alphabet: Finite, nonempty set of symbols e.g.,
  - $\Sigma = \{0, 1\}$: binary alphabet
  - $\Sigma = \{a, b, c, ..., z\}$: the set of all lower case letters
- String: Finite sequence of symbols from an alphabet Σ
  - e.g., 01101 where $\Sigma = \{0,1\}$
  - abracadabra where $\Sigma = \{a, b, c, ..., z\}$

Strings and their properties

- Empty String:
  - The string with zero occurrences of symbols from Σ and is denoted by ε or λ
- Length of String:
  - Number of symbols in the string
  - length of w is indicated by $|w|$
    - $|1010| = 4$
    - $|\varepsilon| = 0$
    - $|uv|=|u|+|v|$
- Reverse :
  - denoted by w^R
  - If w=abc, w^R =cba

Strings and their properties

- Concatenation:
  - if x and y are strings, then xy is the string obtained by placing a copy of y immediately after a copy of x
  - $x=a_1a_2 \ldots a_i$, $y=b_1b_2 \ldots b_j$
  - $xy = a_1a_2 \ldots a_ib_1b_2 \ldots b_j$
  - e.g., x = 01101, y =110, xy = 01101110
  - $x\varepsilon = \varepsilon x = x$

Strings and their properties

- Substring:
  - any string of consecutive characters in some string w
  - if w=abc, then ε, a, ab, abc are substrings of w
- Prefix and suffix:
  - if w = vu
    - v is a prefix of w
    - u is a suffix of w
  - e.g., If w = abc
    - a, ab , abc are prefixes of w
    - c, bc, abc are suffixes of w
    - ε is both prefix and suffix

Strings and their properties

- Power of an Alphabet: Σ^k = the set of strings of length k with symbols from Σ
- Eg.
  - $\Sigma = \{0, 1\}$
    - $\Sigma^0 = \{\varepsilon\}$
    - $\Sigma^1 = \Sigma = \{0,1\}$
    - $\Sigma^2 = \{00, 01, 10, 11\}$

- How many strings are there in Σ^3 ?

Strings and their properties

- The set of all strings over Σ is denoted by Σ^*
  - $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots$
  - Also
    - $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \ldots$
    - $\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$
    - $\Sigma^+ = \Sigma^* - \{\varepsilon\}$

Languages

- Language:
  - set of strings chosen from some alphabet
  - A language is a subset of S*
  - e.g.,
    - The set of valid English words
    - The set of strings consisting of n 0's followed by n 1's {e, 01, 0011, 000111, ...}
    - The set of strings with equal number of 0's and 1's

{e, 01, 10, 0011, 0101, 1010, 1001, 1100, ...}

  - Empty language: $\phi = \{ \}$
    - The language $\{\varepsilon\}$ consists of the empty string
  - Languages can be finite or infinite
  - L = {a, aba, bba} L= $\{a^n | n>0\}$

Operations on Languages

- Complementation
- Reversal
- Union
- Intersection
- Concatenation
- Closure – Kleene and positive

Complementation $\overline{L}$ (or L’)

- Let L be a language over an alphabet Σ.
- The complementation of L
- $\overline{L}$(or L’) = Σ*– L
- Example:
  - Let Σ = {0, 1} be the alphabet.
  - L = {ω ϵ Σ* | the number of 1’s in w is even}
  - L’ = {ω ϵ Σ* | the number of 1’s in w is not even}
  - L’ = {ω ϵ Σ* | the number of 1’s in w is odd}

Reversal (L^R)

- Let L be a language over an alphabet Σ.
- The reversal of L

  $L^R = \{w^R \mid w \text{ is in } L\}$.

- Example
  - $L = \{x \in \{0,1\}^* \mid x \text{ begins with } 0\}$
    - $L^R = \{x \in \{0,1\}^* \mid x \text{ ends with } 0\}$
  - $L = \{x \in \{0,1\}^* \mid x \text{ has } 00 \text{ as a substring}\}$
    - $L^R = \{x \in \{0,1\}^* \mid x \text{ has } 00 \text{ as a substring}\}$

Reversal (L^R)

- $L = \{0,01,011,0111,0110\}$
  - $L^R = \{0,10,110,1110,0110\}$
- $L = \{00,001,01001,011001,00110\}$
  - $L^R = \{00,100,10010,100110,01100\}$

Union

- Let L_1 and L_2 be languages over an alphabet Σ.
- The union of L_1 and L_2 is:
- $L_1 \cup L_2 = \{x \mid x \text{ is in } L_1 \text{ or } L_2\}$.
- Example:
  - $L_1 = \{ x \in \{0,1\}^* \mid x \text{ begins with } 0 \}$
  - $L_2 = \{x \in \{0,1\}^* \mid x \text{ ends with } 0\}$
  - $L_1 \cup L_2 = \{x \in \{0,1\}^* \mid x \text{ begins or ends with } 0\}$

Union

- $\{0,01,011,010,0110\} \cup \{0,10,110,0110\}$
  $=\{0,01,011,010,10,110,0110\}$

Intersection

- Let L_1 and L_2 be languages over an alphabet Σ.
- Intersection of L_1 and L_2 is:
- $L_1 \cap L_2 = \{ x \mid x \text{ is in } L_1 \text{ and } L_2 \}$
- Example:
  - $L_1 = \{ x \in \{0,1\}^* \mid x \text{ begins with } 0\}$
  - $L_2 = \{ x \in \{0,1\}^* \mid x \text{ ends with } 0\}$
  - $L_1 \cap L_2 = \{ x \in \{0,1\}^* \mid x \text{ begins and ends with } 0\}$

Intersection

- $\{0,01,011,010,0110\} \cap \{0,10,110,0110\} = \{0,0110\}$

Concatenation

- Let L_1 and L_2 be languages over an alphabet Σ.
- The concatenation of L_1 and L_2
- $L_1 . L_2 = \{w_1 . w_2 \mid w_1 \text{ is in } L_1 \text{ and } w_2 \text{ is in } L_2 \}$.
- Example:
  - $L_1 = \{ x \in \{0,1\}^* \mid x \text{ begins with } 0\}$
  - $L_2 = \{ x \in \{0,1\}^* \mid x \text{ ends with } 0\}$
    - $L_1 . L_2 = \{ x \in \{0,1\}^* \mid x \text{ begins and ends with 0 and length}(x) \geq 2\}$
    - $L_2 . L_1 = \{ x \in \{0,1\}^* \mid x \text{ has 00 as a substring}\}$

Concatenation

- L^n = L concatenated with itself n times
- $L^0 = \{e\}$; $L^1 = L$; $L^2 = LL$

- $L_1 = \{a, ab\}$, $L_2 = \{bb, b\}$
- $L_1L_2 = \{abb, ab, abbb\}$

Concatenation

- {0,01,010,0110} . {0,0110,110} = {00,010,0100,01100,00110,010110,0100110,01100110,0110,01110,010110,0110110}

- { 0,0110}. {0,01,010,0110} ={00, 001, 0010, 00110, 01100, 011001, 0110010, 01100110}

Kleene's closure (L*) (star closure)

- Let L be a language over an alphabet Σ.
- The Kleene's closure of L
  - $L^* = \{x \mid \text{for an integer } n \geq 0,\ x = x_1\, x_2 \ldots x_n$ and $x_1, x_2, \ldots, x_n$ are in $L\}$.

- $L^* = \cup_{i=0}^{\infty} L^i$
- L * denotes zero or more concatenations of L

Kleene's closure (L^*) (star closure)

- $L=\{a,ab\}$
- $L^* = L^0 \cup L^1 \cup L^2 \cup \ldots$
- $L^* = \{e\} \cup \{a, ab\} \cup \{aa, aab, aba, abab\} \cup \ldots$
- Example:
  - Let $\Sigma = \{0,1\}$ and
  - $L_e = \{w \in \Sigma^* \mid \text{the number of 1's in } w \text{ is even}\}$
  - $L_e^* = \{w \in \Sigma^* \mid \text{the number of 1's in } w \text{ is even}\}$
  - $(L_e')^* = \{w \in \Sigma^* \mid \text{the number of 1's in } w \text{ is odd}\}^*$

    $= \{w \in \Sigma^* \mid \text{the number of 1's in } w > 0\}$

Kleene's closure (L^*) (star closure)

- $L_e = \{110, 1010110, 101\}$
- $L_e^* = \{e\} \cup \{110, 1010110, 101\} \cup \{110110, 1101010110, 110101, 1010110110, 10101101010110, 1010110101, 101110, 1011010110, 101101\} \cup \ldots$

Positive Closure

- Positive closure of L (L^+)
- $L^+ = \bigcup_{i=1}^{\infty} L^i$
- L^+ denotes one or more concatenations of L
- $L^+ = L^* - L^0$

Languages

- $L = \{A, B, ..., Z, a, b, ...z\}$ $D = \{1, 2, ..., 9\}$
- LD = the set of letters and digits
- LD = all strings consisting of a letter followed by a digit
- L^2 = the set of all two-letter strings . e.g, Xy
- $L^4 = L^2L^2$ = the set of all four-letter strings
- L^* = {All possible strings of L plus e }
- $L^+ = L^* - \{ e\}$
- D^+ = set of strings of one or more digits
- $L(L \cup D)$ = set of all strings consisting of a letter followed by a letter or a digit
- $L (L \cup D)^*$ = set of all strings consisting of letters and digits beginning with a letter

Languages

- Learn the proof for $L^+ = LL^*$
  - L^+ is defined as $L + LL + LLL + \ldots\ldots$
  - $L^* = e + L + LL + LLL + \ldots\ldots$
  - $LL^* = Le + LL + LLL + \ldots\ldots$
  - Note that $Le = L$, so, it is seen that the infinite expression L^+ and LL^* are same. So, it is proved that $L^+ = LL^*$

Practice questions

- 1. Is the empty language $\phi = \{e\}$?
- 2. Is $L_1L_2 = L_2L_1$? Prove with an example.
- 3. Consider the language L consisting of strings over {a,b} in which each string begins with an a and should have an even length. Indicate whether each of the following are subsets of L or not.
  - i. {aa,ab} ii. {aaaa,aaab,aaba,aabb,abaa,abab,abba,abbb} ii. {baa} iv. {a}
- 4. Consider the language L consisting of strings over {a,b} in which each occurrence of b is immediately preceded by an a. Indicate whether each of the following strings are elements of L or not.
  - i. e ii. a iii. abaab iv. bb v. bab vi. abb
- 5. Let X = {a,b,c} and Y = {abb, ba}. What will be the values of XY, X^0 , X^1 , and X^2.

Practice questions

- 6. Consider the language L consisting of the strings over {a,b} that contain the substring bb.

  L = {a,b}*{bb}{a,b}* = S * {bb}S *

  Identify if the following are elements of L or not.

  - i. bb ii. abb iii. bbb iv. aabb vi. bbabba vii. abab viii. bab ix. b
- 7. Let L be the language that consists of all strings that begin with aa or end with bb

  a) Write the language L.

  b) For each string below, identify the elements of L.

  i. bb ii. abb iii. bbb iv. aabb v. ba vi. bbaaa vii. bbabba viii. abab ix. bab

Practice questions

- 8. Let $L_1 = \{bb\}$ and $L_2 = \{e, bb, bbbb\}$. The languages L_1^* and L_2^* both contain precisely the strings consisting of ____ number of b's.
  - Is e an element of L_1^* and L_2^* ?
- 9. What is the language of all even-length strings over {a,b}
- 10. What is the language of all odd-length strings over {a,b}

Grammar

- A grammar for the English language tells us whether a particular sentence is well-formed or not.
- If the sentence is correct grammatically then that sentence will be the part of grammar, otherwise not.
  - "I am going to school." - valid
  - "I going am to school." - invalid
- Grammar is a set of rules used to define a language.
- It is the structure of the strings in the language.

Definition of a Grammar

- Definition: A grammar is a 4-tuple G = (V, T, P, S), where
  - V is a finite set of symbols called variables/non-terminals (uppercase letters)
  - T is a finite set of symbols called terminals (usually lowercase letters or special characters)
  - S ϵ V is the start symbol
  - P is the set of productions of the form α -> β
  - Note: $V \cap T = \varnothing$

Derivation

- Non-terminals(Variables) - V
- Terminals – T
- Total alphabet - $V \cup T$

- **α**, **β** are strings in $(V \cup T)^*$
- **α** -> **β**, **β** is obtained from **α** in one step
- **α** ->* **β**, **β** is obtained from **α** in more than zero step ->* reflexive transitive closure of ->

Language generated by Grammar

- Let G = (V, T, P, S) be a grammar. Then the language generated by G, denoted by L(G) is the set of all strings that can be derived from a grammar.

  $$L(G) = \{ w \in T^* \mid S \rightarrow^* w\}$$

- If $w \in L(G)$, then the sequence $S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow w_3 \Rightarrow \ldots \Rightarrow w_n \Rightarrow w$ is a “derivation” of the sentence w.
- The strings S, w_1 , w_2 , w_3 ,w_n which contain variables as well as terminals, are called “SENTENTIAL FORMS” of the derivation.

Language generated by Grammar

- Grammars are language generating devices
- Automata are language accepting devices

Grammar (Examples)

- $G = (\{S, A\}, \{a, b\}, P, S)$, where
- P:
  - $S \rightarrow aA$
  - $A \rightarrow b$
- $L(G) = \{ab\}$

Grammar (Examples)

- G = ({S}, {a, b}, P, S), where
- P:
  - $S \rightarrow aS$
  - S -> b
- L(G) = { b, ab, a^2b, a^3b,. . . }
- L(G) = { $a^nb \mid n \geq 0$ }

Think…

- How to generate a string using a grammar?
- In how many steps a string can be generated by applying the productions in a grammar?
- How to identify the language or check for the string generated by a grammar?

Grammars and Languages

- G= ({S}, {a}, P, S), where

  P: $S \rightarrow aS$ (rule 1)

  $S \rightarrow a$ (rule 2)
- $S \rightarrow a$
- $S \rightarrow aS \rightarrow aa$
- $S \rightarrow aS \rightarrow aaS \rightarrow aaa$
  - $L(G) = \{ a^n \mid n \geq 1 \}$

Grammars and Languages

- $G = (\{S\}, \{a\}, P, S)$, where

  $P: S \rightarrow aS$ (rule 1)

  $S \rightarrow a$ (rule 2)

Grammars and Languages

- $G = (\{S\}, \{a\}, P, S)$, where

  $P: S \rightarrow aS$ (rule 1)

  $S \rightarrow a$ (rule 2)

- $S \rightarrow a$
- $S \rightarrow aS \rightarrow aa$
- $S \rightarrow aS \rightarrow aaS \rightarrow aaa$

Grammars and Languages

- G= ({S}, {a}, P, S), where

  P: $S \rightarrow aS$ (rule 1)

  $S \rightarrow a$ (rule 2)

- $S \rightarrow a$
- $S \rightarrow aS \rightarrow aa$
- $S \rightarrow aS \rightarrow aaS \rightarrow aaa$

➢ $L(G) = \{ a^n \mid n \geq 1 \}$

Grammars and Languages

- $G = (\{S\}, \{a, b\}, P, S)$, where
- $P: S \rightarrow aS$ (rule 1)
- $S \rightarrow b$ (rule 2)

Grammars and Languages

- $G = (\{S\}, \{a, b\}, P, S)$, where
- $P: S \rightarrow aS$ (rule 1)
- $S \rightarrow b$ (rule 2)

- $S \rightarrow b$
- $S \rightarrow aS \rightarrow ab$
- $S \rightarrow aS \rightarrow aaS \rightarrow aab$

Grammars and Languages

- G = ({S}, {a, b}, P, S), where
- P: $S \rightarrow aS$ (rule 1)
- $S \rightarrow b$ (rule 2)

- $S \rightarrow b$
- $S \rightarrow aS \rightarrow ab$
- $S \rightarrow aS \rightarrow aaS \rightarrow aab$

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

Grammars and Languages

- $G = (\{S\}, \{a, b\}, P, S)$, where
- $P: S \rightarrow aSb$ (rule 1)
- $S \rightarrow ab$ (rule 2)

Grammars and Languages

- G= ({S}, {a, b }, P, S), where
- P: S → aSb (rule 1)
- S → ab (rule 2)

- $L(G) = \{ a^n b^n \mid n \geq 1 \}$

Grammars and Languages

- $G = (\{S\}, \{a, b, c\}, P, S)$, where
- $P: S \rightarrow aSa$ (rule 1)
- $S \rightarrow bSb$ (rule 2)
- $S \rightarrow c$ (rule 3)

Grammars and Languages

- $G = (\{S\}, \{a, b, c\}, P, S)$, where
- $P: S \rightarrow aSa$ (rule 1)
- $S \rightarrow bSb$ (rule 2)
- $S \rightarrow c$ (rule 3)

$$L(G) = \{ wcw^R \mid w \in \{a, b\}^* \}$$

Grammars and Languages

- G= ({S,B}, {a, b, c }, P, S), where
- P: S $\rightarrow$ aSBc (rule 1)
- S $\rightarrow$ abc (rule 2)
- cB $\rightarrow$ Bc (rule 3)
- bB $\rightarrow$bb (rule 4)

Grammars and Languages

- $G = (\{S,B\}, \{a, b, c\}, P, S)$, where
- P: $S \rightarrow aSBc$ (rule 1)
- $S \rightarrow abc$ (rule 2)
- $cB \rightarrow Bc$ (rule 3)
- $bB \rightarrow bb$ (rule 4)

- $L(G) = \{ a^n b^n c^n \mid n \geq 1 \}$

Derivation – leftmost and rightmost

- Leftmost derivation − A leftmost derivation is obtained by applying production to the leftmost variable in each step.
- Rightmost derivation − A rightmost derivation is obtained by applying production to the rightmost variable in each step.

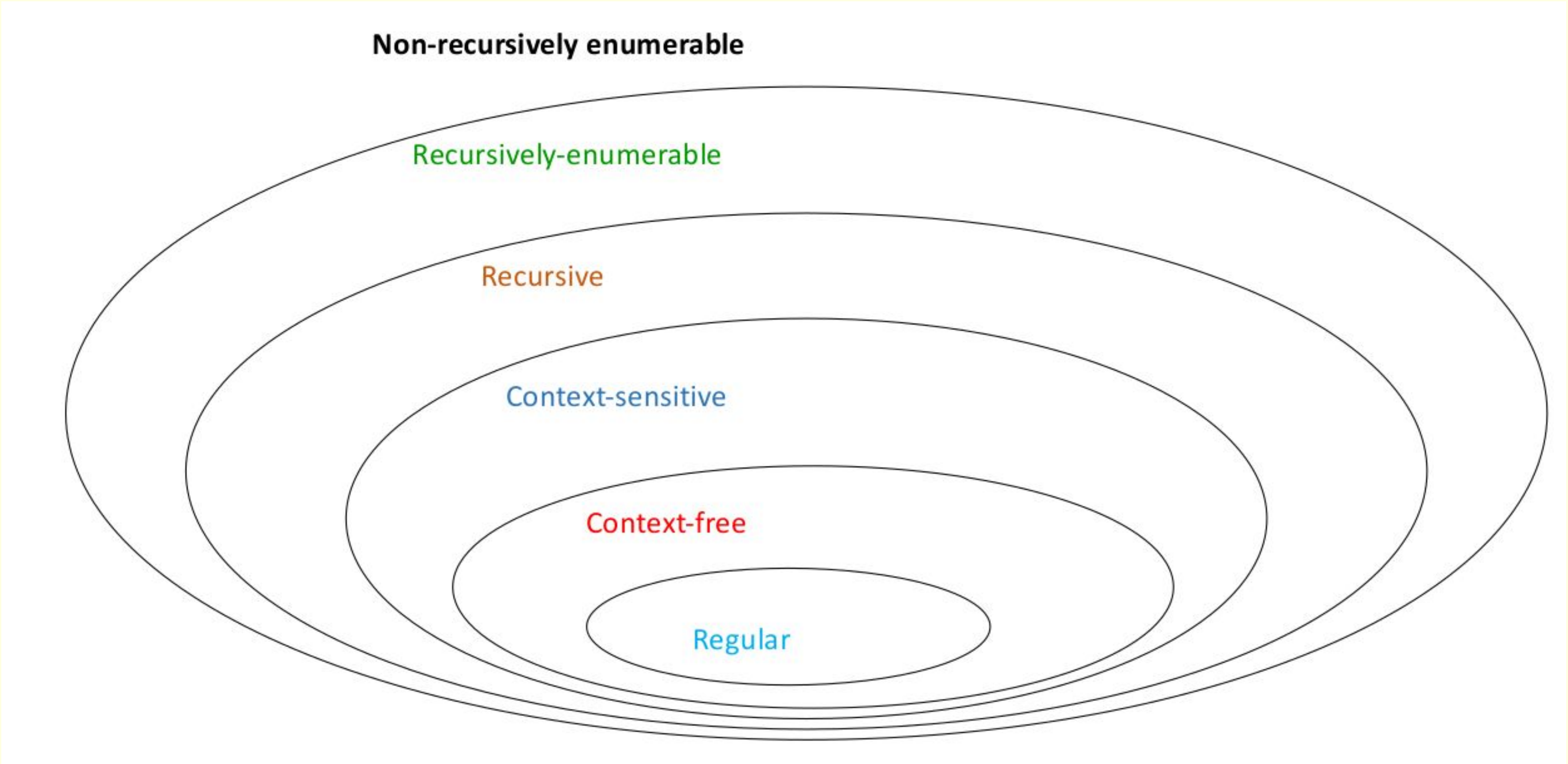
Derivation – leftmost and rightmost

- G= ({E}, {+, *, (,), id }, P, E), where
- P: $E \rightarrow E + E$ (rule 1)
- $E \rightarrow E * E$ (rule 2)
- $E \rightarrow (E)$ (rule 3)
- $E \rightarrow id$ (rule 4)
- Leftmost derivation
  - $E \rightarrow E + E \rightarrow id + E \rightarrow id + E * E \rightarrow id + id * E \rightarrow id + id * id$
- Rightmost derivation
  - $E \rightarrow E + E \rightarrow E + E * E \rightarrow E + E * id \rightarrow E + id * id \rightarrow id + id * id$

Types of grammars

Type	Language	Grammer	Machine
Type- 3	Regular languages	Regular grammars • Right-linear grammars • Left-linear grammars	Finite-state automata
Type - 2	Context-free languages	Context-free grammars	Push-down automata
Type- 1	Context-sensitive languages	Context-sensitive grammars	Linear-bound automata
Type- 0	Recursive languages Recursively enumerable languages	Unrestricted grammars	Turing machines

Chomsky Hierarchy



Thank You