Parul®
University

THEORY OF COMPUTATION
CODE:303105306

UNIT 3 PART II:CLOSURE PROPERTY OF CFL

CHAPTER-3

Closure Properties of CFL

Closure properties of CFLs

- **CFLs are closed under**
  - Union
  - Concatenation
  - Kleene Star
  - Reversal
  - Homomorphism
- **Regular Languages closed under**
  - Intersection
  - Difference
  - Complementation
- **CFLs are not closed under intersection, difference, or complementation, why?**

Parul® University

DIGITAL LEARNING CONTENT

Union

- If L1 and L2 are two context free languages, their union L1 U L2 will also be context free language.
- Let L1 and L2 be CFLs with grammars G1 and G2, respectively.
- Assume G1 and G2 have no variables in common.
- Let S1 and S2 be the start symbols of grammar G1 and G2.
- Form a grammar for (L1 U L2) by combining all the symbols and productions of G1 and G2.
- Add a new start symbol S and productions S -> S1 | S2 in new grammar.

Union

- **Example:**

  $L1 = \{ a^n b^n c^m \mid n,m >= 0 \}$

  $L2 = \{ a^n b^m c^m \mid n,m >= 0 \}$

  $L3 = L1 \cup L2 = \{ a^n b^n c^m \cup a^n b^m c^m \mid n, m >= 0 \}$ is also context free.

- Here Language L1 generates all strings that contains occurrence of a's equals to occurrence of b's and L2 generates all strings that contains occurrence of b's equals to the occurrence of c's.

- Union require either of two condition require to be true. It can be accepted by pushdown automata.

- So, Language L3 is also CFL.

Concatenation

- L1 and L2 are CFLs, then their concatenation L1.L2 will also be context free
- Let L1 and L2 be CFL's with grammars G1 and G2, respectively.
- Assume G1 and G2 have no variables in common.
- Let S1 and S2 be the start symbols of grammar G1 and G2.
- Form a new grammar for L1.L2 by starting with all symbols and productions of G1 and G2.
- Add a new start symbol S and production S -> S1S2.
- Every derivation from S results in a string in L1 followed by one in L2.

Concatenation

- **Example :**

  $L1 = \{ a^n b^n \mid n >= 0 \}$

  $L2 = \{ c^m d^m \mid m >= 0 \}$

  $L3 = L1.L2 = \{a^n b^n c^m d^m \mid n,m >= 0\}$ is also context free language.

- Here L1 grammar generate all strings which contains equal occurrence of a's and b's and L2 generate equal occurrence of c's and d's. Language L3 can be accepted by Pushdown automata.
- Hence, Concatenation is closed under CFLs.

Kleene Star

- L1 is context free, then its Kleene closure L1* will also be context free.
- Let language L have a grammar G with start symbol S1.
- Grammar for L* by adding a new start symbol S and productions S -> S1S | ε to G.
- A rightmost derivation from S generates a sequence of 0's or more S1's, each of which generates some string w in L.
- **Example :**

  $L1 = \{ a^n b^n \mid n >= 0 \}$

  $L1^* = \{ a^n b^n \mid n >= 0 \}^*$ is also context free language.

Reversal

- L is a Context-free language with grammar G
- Grammar for L_R by reversing the right side of every production.
- **Example:**

  Let grammar G have S -> 0S1 | 01.

  S -> 1S0 | 10.

- L_R is also Context-free grammar.

Homomorphism

- Let language L be a context-free language with grammar G.
- Let h be a homomorphism on the terminal symbols of grammar G.
- Construct a grammar G’ for h(L) by replacing each terminal symbol a by h(a).
- **Example:**

Grammar G has productions S -> 0S1 | 01.

Homomorphism h is defined by h(0) = ab, h(1) = ε.

h(L(G)) has the grammar G’ with productions S -> abS | ab.

Non-Closure Properties of CFL's - Intersection

- Let L1 and L2 are two context free languages

  $L1 = \{ a^n b^n c^m \mid n, m >= 0 \}$

  $L2 = (a^m b^n c^n \mid n, m >= 0 \}$

  $L3 = L1 \cap L2 = \{ a^n b^n c^n \mid n >= 0 \}$

- L1 generate all strings of number of a's should be equal to number of b's and

  L2 generate all strings of number of b's should be equal to number of c's.
- Intersection require both conditions need to be true
- It cannot be accepted by pushdown automata, so it is not context free.

Complementation

- Let L1 and L2 are two context free languages

  $L1 = \{ a^n b^n c^m \mid n, m >= 0 \}$

  $L2 = (a^m b^n c^n \mid n, m >= 0 \}$

  $(L1' \cup L2')' = L1 \cap L2 = \{a^n b^n c^n \mid n >= 0 \}$

- Context-free languages(CFLs) are not closed under intersection property.
- Language is not context free and it can not accepted by Pushdown automata.
- Hence, Context-free languages (CFLs) are not closed under Complementation.

Difference

- Let L1 and L2 are two context free languages
- Proof: $L1 \cap L2 = L1 - (L1 - L2)$.
- Context-free languages(CFLs) are not closed under Intersection property.
- If CFLs were closed under difference, they require to be closed under intersection, but they are not.
- Thus, Context-free languages(CFLs) are not closed under difference

Context Sensitive Grammar and Languages

- **Hierarchy of languages.**

  - Type-0 : Recursively enumerable language
  - Type-1 : Context Sensitive language
  - Type-2 : Context Free language
  - Type-3 : Regular language

- Brief discussion on Context Sensitive Language and corresponding state machine, (Linear Bounded Automaton(LBA)), equivalence and properties of Context Sensitive Languages.

Definition : Context Sensitive Grammar(CSG)

- Context Sensitive Grammar (CSG) is a quadruple G=(N,∑,P,S) where,
  - N is set of non-terminal symbols
  - ∑ is set of terminal symbols
  - S is start symbol
  - P is set of production in form of $\alpha A \beta \rightarrow \alpha \gamma \beta$ where $\gamma \neq \epsilon$
- Derivation non-terminal A will be changed to γ only
- CSG is Non-contracting grammar as $\gamma \neq \epsilon$ then $\alpha \rightarrow \beta \Rightarrow |\alpha| \leq |\beta|$

Context Sensitive Language(CSL)

- The language that can be defined by context-sensitive grammar is called Context sensitive language(CSL).

- **Example:**

  Consider the following CSG.

  S → abc/aAbc

  Ab → bA

  Ac → Bbcc

  bB → Bb

  aB → aa/aaA

- **Derivation of CSL**

  S → aAbc
  → abAc
  → abBbcc
  → aBbbcc
  → aaAbbcc
  → aabAbcc
  → aabbAcc
  → aabbBbccc
  → aabBbbccc
  → aaBbbbccc
  → aaabbbccc

  Context sensitive language L= $\{a^n b^n c^n \mid n \geq 1\}$.

Closure properties of CSLs

- Union
- Intersection
- Complement
- Concatenation
- Kleene closure
- Reversal

Equivalence of CSL

- The following grammar(G) is context-sensitive.

$$S \rightarrow aTb \mid ab$$
$$aT \rightarrow aaTb \mid ac$$

- Language L(G) generated by grammar G

$$L(G) = \{ab\} \cup \{a^n c b^n \mid n>0\}$$

- Language is also a context-free.
- Example: Context free grammar(G1) for language L(G).

$$S \rightarrow aTb \mid ab$$
$$T \rightarrow aTb \mid c$$

- Any context-free language is context sensitive language.
- Not all context sensitive but it need not be context free.

Equivalence of CSL

Theorem: Every context-sensitive language L is recursive.

- Let L be CSL, G be CSG
- Derivation of string w, S ->S1 ,S1->S2, S2->S3....... = w
- No of steps are Bound on possible derivations. We know that |xi|< |xi +1|

  (G is non-contracting).
- Check whether string w is in L(G) as follows

  -Construct a transition graph where vertices are the strings of length|w|.

  -To find Paths correspond to derivation in G. Add edge from x to y if x -> y

  -w ∈ L(G) if there is a path from S to w.

Equivalence of CSL

Theorem: There exists some recursive language that is not context sensitive language.

- **Language L is recursive**

  -Create context-sensitive grammar Gi = (Ni , {0; 1; 2; 3; 4; 5; 6; 7; 8; 9},Si , Pi) which generates numbers.

  -We can define language L, which contains the numbers of the grammars which doesn’t generate the number of its position in the list.

$$L = \{i \mid i \notin L(G_i)\}.$$

  -We can create a list of context-sensitive generative grammars which generates numbers, and we can decide whether a context-sensitive grammar generates its position in the list or not.

  -Hence, language L is recursive.

Equivalence of CSL

Theorem: There exists a recursive language that is not context sensitive.

- **Language L is not context sensitive language**

  -Assume that L is a CSL

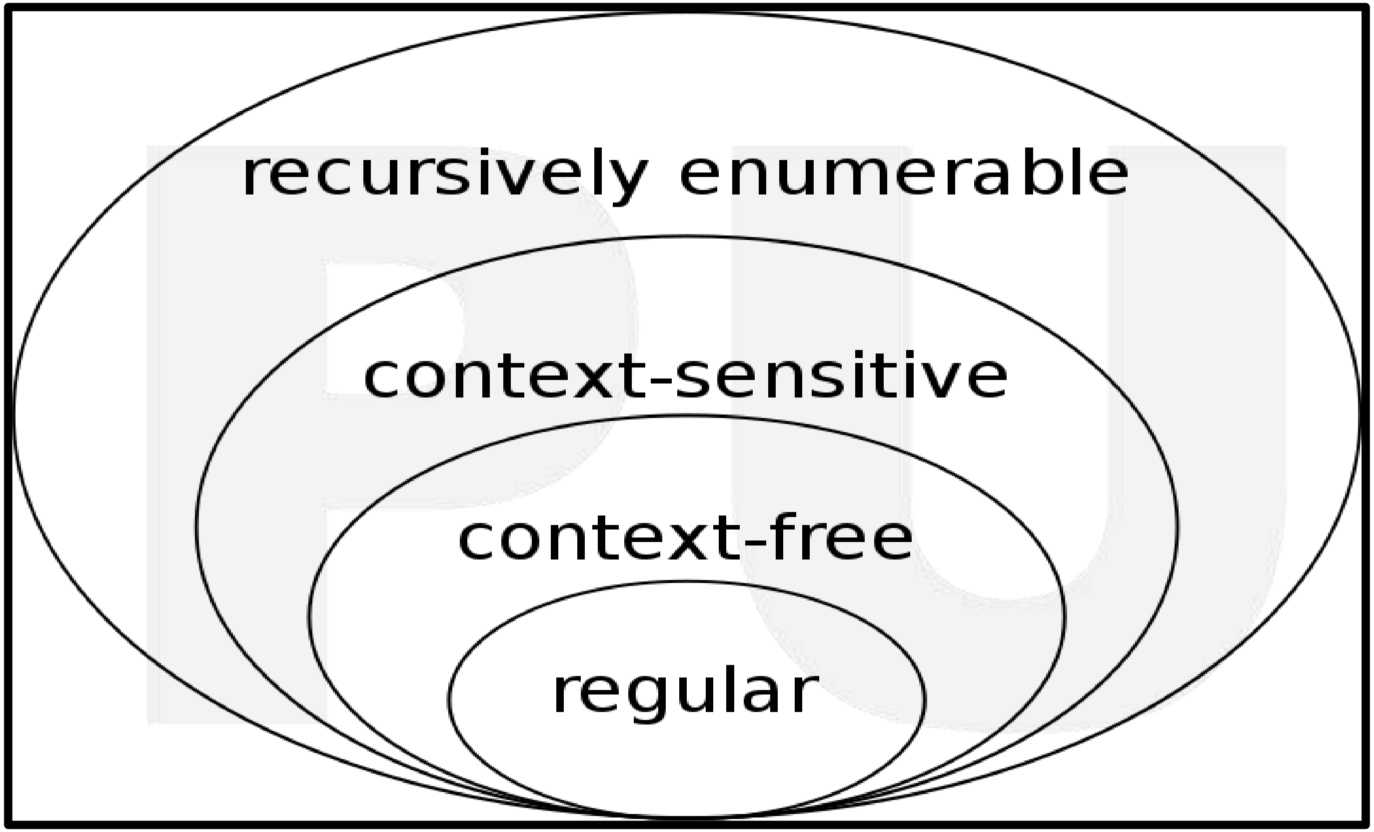
  -So there is a CSG Gc , s.t L(Gc) = L for some c.

  -If c ∈ L(Gc), by the denition of L, we have c ∉ L, but L = L(Gc). So a contradiction.

  -If c ∉ L(Gc), then c ∈ L is also a contradiction since L = L(Gc).

  -Hence, language L is not context sensitive language.

Chomsky Hierarchy



Chomsky Hierarchy

Type	Language	Automaton	Production rules
Type-0 Unrestricted	Recursively enumerable	Turing Machine	$\alpha \rightarrow \beta$
Type-1 Context-sensitive	Context-sensitive language	Linear-bounded automaton	$\alpha A\beta \rightarrow \alpha\gamma\beta$
Type-2 Context-free	Context-free language	Pushdown automaton	$A \rightarrow \gamma$
Type-3 Regular	Regular language	Finite state automaton	$A \rightarrow a$ And $A \rightarrow aB$

Linear Bounded Automata

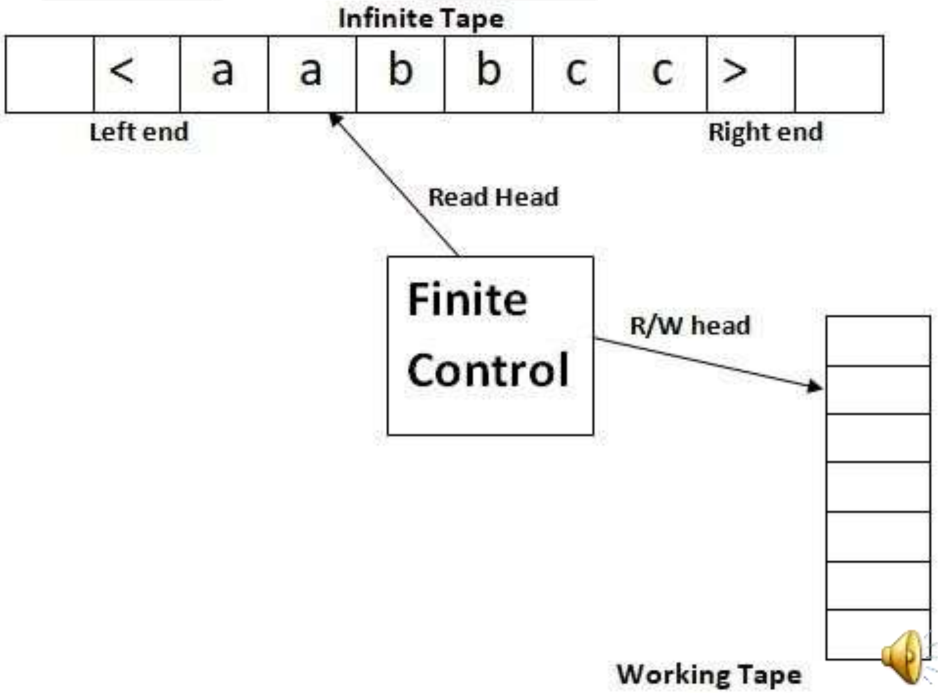
- **Linear Bounded Automata is a single tape non-deterministic Turing Machine with two special tape symbols left marker ‘< ‘ and right marker ‘>’.**
- **The transitions should satisfy below conditions:**
  - **It should not replace any other symbol in place of marker symbols..**
  - **It should not write on tape cell beyond the marker symbols.**
- **Configuration of string will be: < q0a1a2a3a4a5…….an > = <q0w>**

Linear Bounded Automata

- **Linear Bounded Automata is a single tape non-deterministic Turing Machine,**

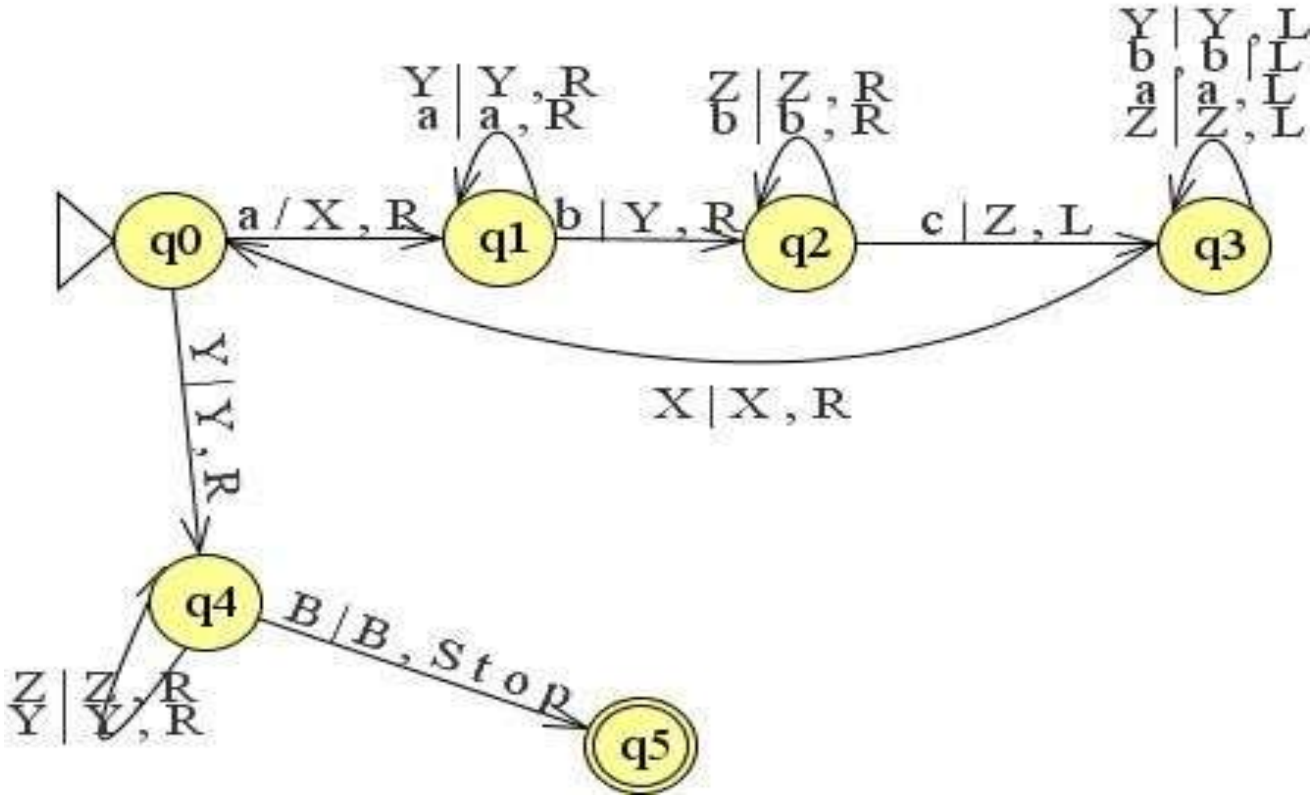
  M=(Q, Σ ,T, δ, B, ,F, q0, <, >, t, r) Where,

  - Q is set of states
  - Σ is set of terminals
  - T is set of tape alphabets
  - δ is set of transitions
  - B is blank symbol
  - q0 is the initial state
  - < is left marker symbol
  - > is right marker symbol
  - t is accept state
  - r is reject state



Linear Bounded Automata

- **Turing Machine for Context sensitive language L= {$a^n b^n c^n$ | n≥1}.**



DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in