15/04/21

# FA
## TUTORIAL - 6

JUNAID . GIRKAR
60004190057
SE COMPS A-3

Q1 Define Pushdown Automata.

ANS Pushdown Automata is a finite automata with extra memory called stack which helps Pushdown automata to recognize Context Free Languages.

A Pushdown Automata (PDA) can be defined as:
- → $Q$ is the set of states
- → $\Sigma$ is the set of input symbols
- → $\Gamma$ is the set of pushdown symbols.
- → $q_0$ is the initial state.
- → $Z$ is the initial pushdown symbol
- → $\delta$ is a transition function which maps $Q \times \{\Sigma \cup \epsilon\} \times \Gamma$ into $Q \times \Gamma^*$. In a given state, PDA will read input symbol and stack symbol and move to a new state and change the symbol of stack.

Q2 Give a PDA to accept the language $L = \{0^n 1^m$ wher $n \leq m\}$

ANS STEP 1: Defination $M = (Q, \Sigma, \Gamma, \delta, Z_0, F, q_0)$

where, $Q$ = set of states
$\Sigma$ = set of symbols
$\Gamma$ = set of pushdown symbols
$q_0$ = initial state
$Z_0$ = initial pushdown symbol
$F$ = set of Final states
$\delta$ = Transition $Q \times \{\Sigma \cup \epsilon\} \times \Gamma \rightarrow Q \times \Gamma^*$

$L = 0^n 1^m$ , $n <= m$

$= 0^n 1^n . 1^x$ $m = n + x$

STEP 2: logic

push X for each 0 in stack

pop X for each 1 in stack

STEP 3: Description

$Q = \{q_0, q_1, q_f\}$

$\Sigma = \{0, 1\}$

$\Gamma = \{Z_0, X\}$

$\delta = Q \times \{\Sigma \cup \epsilon\} \times \Gamma \longrightarrow Q \times \Gamma^*$

$Z_0 = Z_0$

$F = q_f$

$q_0 = q_0$ (Initial state)

STEP 4: Transition

$\delta(q_0, 0, Z_0) \longrightarrow (q_0, XZ_0)$

$\delta(q_0, 0, X) \longrightarrow (q_0, XX)$

$\delta(q_0, 1, X) \longrightarrow (q_1, \epsilon)$

$\delta(q_1, 1, X) \longrightarrow (q_1, \epsilon)$

$\delta(q_1, 1, Z_0) \longrightarrow (q_1, Z_0)$

$\delta(q_1, \epsilon, Z_0) \longrightarrow (q_f, Z_0)$

Q.3 Give a PDA that accept the language $L = \{a^i b^j c^k$ where $i, j, k \geq 0$ and $i + j = k\}$

ANS

STEP 1: Defination

$M = (Q, \Sigma, \Gamma, \delta, Z_0, F, q_0)$

where $Q$ = set of states

$\Gamma$ = set of pushdown symbols

$q_0$ = Initial state

$Z_0$ = Initial pushdown symbol

$F$ = set of final states.

$\delta$ = Transition

$Q \times \{\Sigma \cup \epsilon\} \times \Gamma \longrightarrow Q \times \Gamma^*$

$L = a^i b^j c^k$ , $i, j, k \geq 0$ , $i + j = k$

$L = a^i b^j c^j c^i$

STEP 2: LOGIC

1. For every input symbol 'a', a symbol X is pushed onto the stack.
2. For every input symbol 'b', a symbol X is pushed onto the stack
3. For every input symbol 'c', X is erased from the stack

STEP 3: DESCRIPTION

$Q = \{q_0, q_1, q_2, q_f\}$

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

$\Gamma = \{Z_0, X\}$

$\delta = Q \times \{\Sigma \cup \epsilon\} \times \Gamma \longrightarrow Q \times \Gamma^*$

$Z_0 = Z_0$

$F = q_f$

$q_0 = q_0$ (Initial state)

STEP 4: TRANSITION

$\delta(q_0, a, Z_0) \longrightarrow (q_0, XZ_0)$
$\delta(q_0, a, X) \longrightarrow (q_0, XX)$
$\delta(q_0, b, Z_0) \longrightarrow (q_1, XZ_0)$
$\delta(q_0, b, X) \longrightarrow (q_1, XX)$
$\delta(q_1, b, X) \longrightarrow (q_1, XX)$
$\delta(q_1, c, X) \longrightarrow (q_2, \epsilon)$
$\delta(q_2, c, X) \longrightarrow (q_2, \epsilon)$
$\delta(q_2, \epsilon, Z_0) \longrightarrow (q_f, Z_0)$
$\delta(q_0, \epsilon, Z_0) \longrightarrow (q_f, Z_0)$

Q4 Write down properties of content free language.

ANS Properties of content Free language (CFL):

- A content free language can be recognized by a PDA
- For every CFL, there exists a PDA.
- The language of PDA [Pushdown Automata] is a CFL.
- The content free languages are closed under some specific operation, closed means [redacted] after doing that operation on a CFL the resultant language will also be a CFL. [closure property]

Some such operation are :-

→ Union operation
→ Concatenation
→ Kleene closure
→ Reversal operation
→ Homomorphism
→ Inverse Homomorphism
→ Substitution
→ Init (or) prefix operation.
→ Quotient with regular language
→ cycle operation
→ Union with regular language.
→ Intersection with regular language
→ Difference with regular language.

CFL is not closed under some specific operation, not-closed means after doing that operation on a context free language the resultant language is no longer a CFL.

Some such operation are :-

- Intersection
- complement
- subset
- superset
- Infinite Union
- Difference, symmetric difference (XOR, NAND, NOR or any other operation which get reduced to intersection and complement)

- Decision Properties :
  1. Test for membership : Decidable
  2. Test for emptiness : Decidable
  3. Test for finiteness : Decidable

  Rest of the decision properties are undecidable in CFL.

- Deterministic Property :

  The CFL can be :-

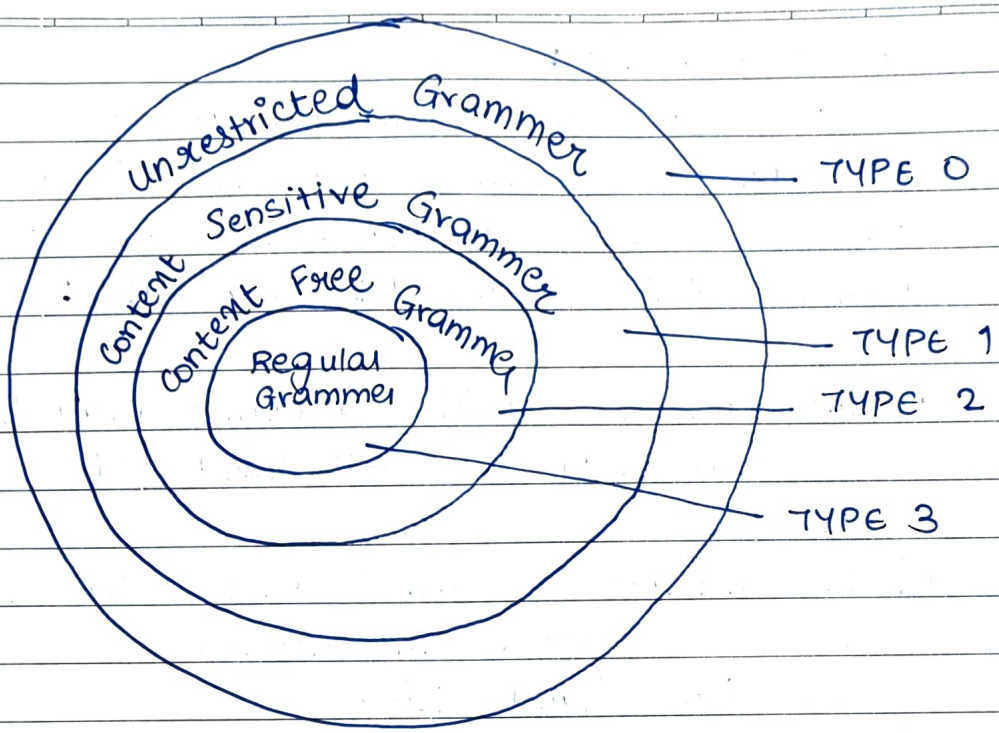
  → DCFL - Deterministic [which can be recognized by Deterministic Pushdown Automata] CFL.

  → NDCFL - Non-deterministic [can't be recognized by DPDA but NDPA] CFL

Q5 Explain chomsky hierarchy of languages.

ANS Chomsky Hierarchy represents the class of languages that are accepted by the different machine.
The category of language in chomsky's Hierarchy is as given below :-

1) Type 0 Known as unrestricted Grammer.
2) Type 1 Known as content sensitive Grammer.
3) Type 2 Known as content free grammer.
4) Type 3 Regular Grammer.




Chomsky Hierarchy.

This is hierarchy. Therefore every language of type 3 is also of types 2, 1, 0. Similarly, every language of type 2 is also of type 1 and type 0.

Type 0 Grammer :

Type 0 grammer is known as unrestricted grammer. There is no restriction on the grammer rules of these types of languages. These languages can be efficiently modeled by Turing machines.

For e.g : $bAa \longrightarrow aa$

$S \longrightarrow s$

Type 1 Grammer :

Type 1 grammer is known as content sensitive grammer. The content sensitive grammer is used to represent content sensitive language. The content sensitive grammer follows the following rules:

a] The content sensitive grammer may have more than one symbol on the left hand side of their production rules.

b] The number of symbols on the left-hand-side must not exceed the number of symbols on the right hand side.

c] The rule of the form $A \rightarrow \epsilon$ is not allowed unless A is a start symbol. It does not occur on the right-hand side of any rule.

d] The type 1 grammer should be type 0. In type 1, production is in the form of $V \rightarrow T$

where the count of symbol in V is less than or equal to T.

For e.g :

$S \rightarrow AT$

$T \rightarrow xy$

$A \rightarrow a$

Type 2 Grammer :

Type 2 grammer is known as content free grammer. Content free languages are the languages which can be represented by the content free grammer (CFG). Type 2 should be type 1. The production rule is ~~the~~ of the form.

$$A \longrightarrow \alpha$$

where A = any single non-terminal / variable

$\alpha$ = any combination of variable and terminal.

For e.g. $A \longrightarrow aBb$

$A \longrightarrow b$

$B \longrightarrow a$

Type 3 Grammer :

Type 3 grammer is known as Regular Grammer. Regular languages are those languages which can be described using regular expressions.

These languages can be modeled by NFA or DFA

Type 3 is most restricted form of grammer. The type 3 grammer should be type 2 and type 1.

Type 3 should be in the form of :

$$V \longrightarrow T^*V / T^*$$

For e.g.

$A \longrightarrow xy$