FLAT Tutorial No :- 6

(Based on Module 3)

PRN : 2020BTECS00212

Sub : FLAT

Subject Code : 5CS222

Date : 14/05/2021

… … … … … … … … … … … … … ..

Q1. Define Push Down Automata.

Q2. Construct a PDA that accepts the language = $\{a^n b^{4n} | n>=1\}$, where ∑ = {a, b}

Supported strings like L={abbbb, aabbbbbbbb, aaabbbbbbbbbbbb,… … }

Q3. Construct a PDA that accepts the language = $\{a^n b^n | n>=1\}$, where ∑ = {a, b}

Supported strings like L={ab, aabb, aaabbb,… … }

Answers :

Tutorial No:- 6.

PRN:- 2020BTECS00212
Name:- Pranita Rajendra Bhosale.
Sub:- FLAT.
Source Code:- SC5222
Batch:- S8.
Date:- 14/05/2021.

Que.1. Define Push Down Automata :-

Ans:- Pushdown Automata (PDA) is a way to implement a CFG in the same way we design DFA for a regular grammar. A DFA can remember a finite amount of information, but a PDA can be remember an infinite amount of information.

- Pushdown automata is simply an NFA augmented with an "external stack memory".
- The addition of a stack is used to provide a last-in-first-out memory management capability to pushdown automata.
- It can store an unbounded amount of information on the stack. It can access a limited amount of information on the stack. A PDA can push an element onto the top of the stack & pop off an element from the top of the stack.
- To read an element into the stack, the top elements must be popped off & are lost.

Que.2. Construct a PDA that accepts the language. $\{a^n b^{4n} | n>=1\}$, where, $\Sigma = \{a, b\}$. Supported strings like $L = \{abbbb, aabbbbbbbb, aaabbbbbbbbbbbbb, \ldots\}$.

Ans:- The ID can be Constructed as follows:-

$\delta\{q_0, a, z\} = \{q_0, aaz\}$

$\delta\{q_0, a, a\} = \{q_0, aaa\}$

- Now when we read b, we will change the state from q_0 to q_1 & start popping corresponding 'a'.

Hence,

- $\delta\{q_0, b, a\} = \{q_1, \varepsilon\}$

- Thus this process of popping 'b' will be repeated unless all the symbols are read. Note that popping action occures in state q_1 only.

- $\delta\{q_1, b, a\} = \{q_1, \varepsilon\}$

- After reading all b's all the corresponding a's should get popped. Hence when we read ε as input symbol then there should be nothing in the stack. Hence the move will be:

$\delta\{q_1, \varepsilon, z\} = \{q_2, \varepsilon\}$.

- where,

PDA $= \{\{q_0, q_1, q_2\}, \{a, b\}, \{a, z\}, \delta, q_0, z, \{q_2\}\}$

- we can summarize the ID as:

$\delta\{q_0, a, z\} = \{q_0, aaz\}$

$\delta\{q_0, a, a\} = \{q_0, aaa\}$

$\delta\{q_0, b, a\} = \{q_1, \varepsilon\}$

$\delta\{q_1, b, a\} = \{q_1, \varepsilon\}$

$\delta\{q_1, \varepsilon, z\} = \{q_2, \varepsilon\}$

- Now, we will simulate this PDA for the input string, "aaabbbbbb".

- $\delta\{q_0, aaabbbbbb, Z\}$

$\vdash \delta\{q_0, aabbbbbb, aaZ\}$

$\vdash \delta\{q_0, abbbbbb, aaaaZ\}$

$\vdash \delta\{q_0, bbbbbb, aaaaaaZ\}$

$\vdash \delta\{q_1, bbbbb, aaaaaZ\}$

$\vdash \delta\{q_1, bbbb, aaaaZ\}$

$\vdash \delta\{q_1, bbb, aaaZ\}$

$\vdash \delta\{q_1, bb, aaZ\}$

$\vdash \delta\{q_1, b, aZ\}$

$\vdash \delta\{q_1, \varepsilon, Z\}$

$\vdash \delta\{q_2, \varepsilon\}$

Accept.

Que.3. Construct a PDA that accepts the language = $\{a^n b^n | n>=1\}$, where, $\Sigma = \{a,b\}$, Supported strings like, $L = \{ab, aabb, aaabbb, \ldots\}$.

Ans:- Approach used in the Construction of PDA:-

- As we want to design a NPDA, thus every time 'a' comes before 'b'. When 'a' comes then also push it in stack & if again 'a' comes then also push it.

- After that, when 'b' comes then pop one 'a' from the stack each time, So, at the end if the stack becomes empty then we can say that the string is accepted by the PDA.

- Stack transition functions:-

$\delta(q_0, a, z) \vdash (q_0, az)$
$\delta(q_0, a, a) \vdash (q_0, aa)$
$\delta(q_0, b, a) \vdash (q_1, \epsilon)$
$\delta(q_1, b, a) \vdash (q_1, \epsilon)$
$\delta(q_1, \epsilon, z) \vdash (q_f, z)$

- where,
q_0 = Initial State.
q_f = final State.
ϵ = indicates pop operation.

- So this is our required non-deterministic PDA for accepting the language $L = \{a^n b^n : n>=1\}$.

→ Required PDA is:-

.....END.....