L3 Pushdown Automata Example -3

5.4 The Language of a PDA

SPPU - Dec. 15

University Question

Q. Explain the equivalence of PDA with acceptance by final state and empty stack. **(Dec. 2015, 6 Marks)**

A language L can be accepted by a PDA in two ways :

1. Through final state.
2. Through empty stack.

It is possible to convert between the two classes.

1. From final state to empty stack.
2. From empty stack to final state.

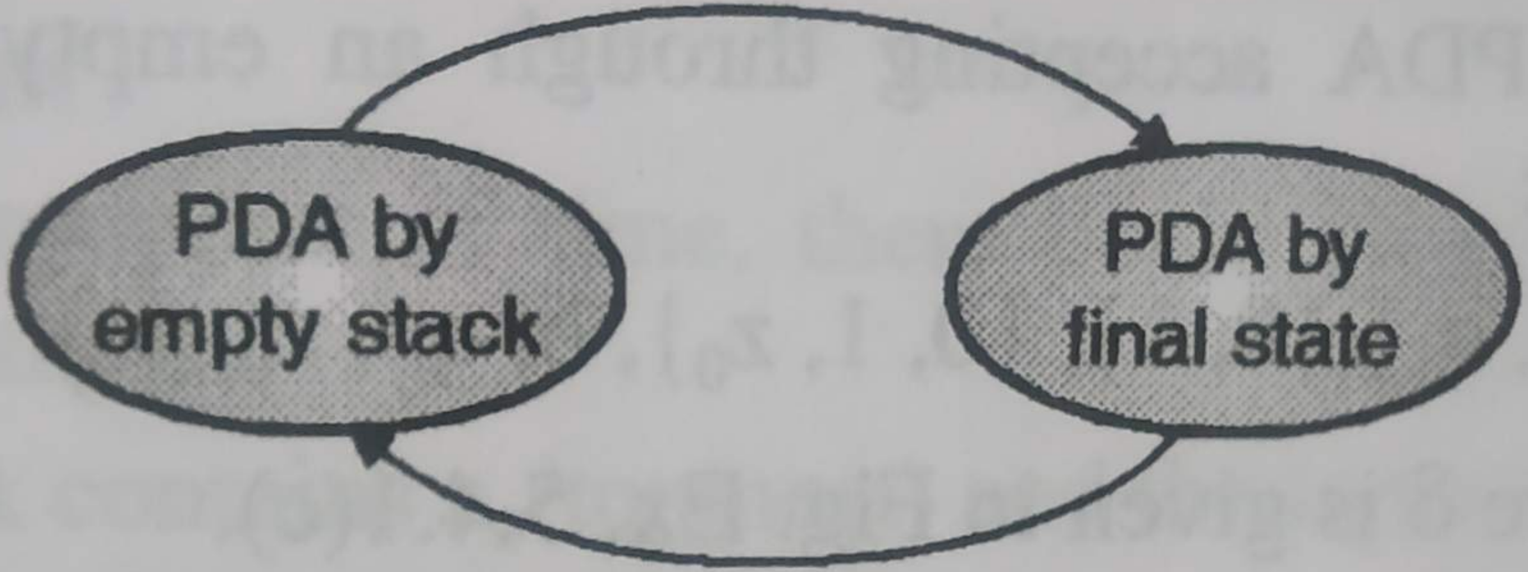

Fig. 5.4.1 : Equivalence of two PDAs

5.4.1 Acceptance by Final State

SPPU - Dec. 13

University Question

Q. Give formal definition of acceptance by PDA in terms of final state. **(Dec. 2013, 2 Marks)**

Let the PDA, $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ then the language accepted by M through a final state is given by :

$$L(M) = \left\{ w \mid (q_0, w, z_0) \overset{*}{\underset{M}{\Rightarrow}} (q_1, \varepsilon, \alpha) \right\}$$

Where the state $q_1 \in F$. α, the final contents of the stack are irrelevant as a string is accepted through a final state.

5.4.2 Acceptance by Empty Stack

SPPU - Dec. 13

University Question

Q. Give formal definition of acceptance by PDA in terms of null store. **(Dec. 2013, 2 Marks)**

Let the PDA, $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, \phi)$ then the language accepted through an empty stack is given by :

$$L(M) = \left\{ w \mid (q_0, w, z_0) \overset{*}{\underset{M}{\Rightarrow}} (q_1, \varepsilon, \varepsilon) \right\}$$

Where q_1 is any state belonging to Q and the stack becomes empty on application of input string w.

Example 5.4.1

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

1. Through empty stack.
2. Through final state.

Solution : Algorithm

1. Sequence of 0's should be pushed onto the stack in state q_0.

$\delta(q_0, 0, z_0) = (q_0, 0z_0)$ [Push the first 0]

$\delta(q_0, 0, 0) = (q_0, 00)$ [Push subsequent 0's]

2. A '0' should be popped for every 1 as input till the stack becomes empty.

$\delta(q_0, 1, 0) = (q_1, \varepsilon)$ [Pop on first 1 and change the state to q_1]

$$\delta(q_1, 1, 0) = (q_1, \varepsilon)$$ [Pop on subsequent 1 as input till every 0 is erased from the stack]

. Subsequent 1's (m – n) will have no effect on the stack.

$$\delta(q_1, 1, z_0) = (q_1, z_0)$$

. Finally, the symbol Z_0 should be popped out to make the stack empty.

$$\delta(q_1, \varepsilon, z_0) = (q_1, \varepsilon)$$

This step is required if the language is to be accepted through an empty stack.

Fig. Ex. 5.4.1.

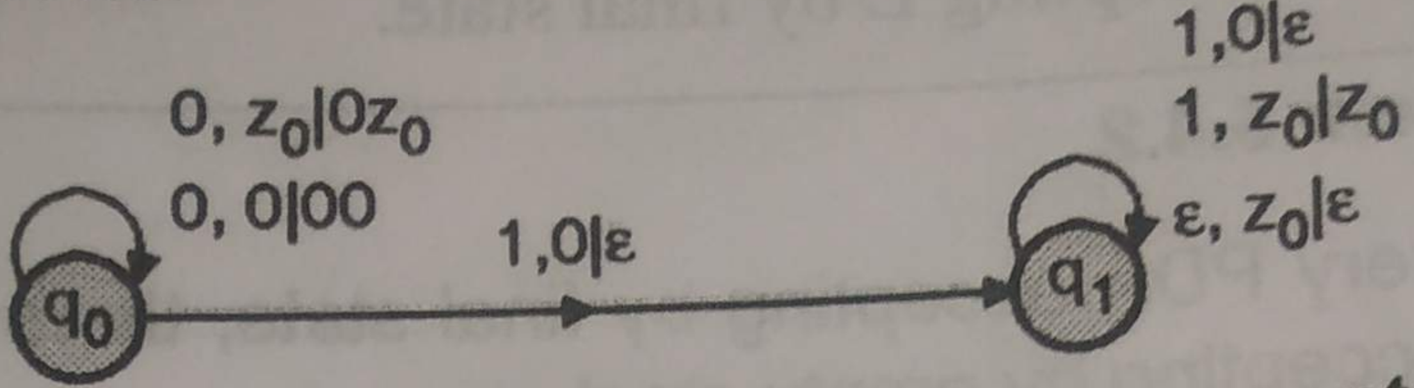

Fig. Ex. 5.4.1(a) : Transition diagram for acceptance through an empty stack

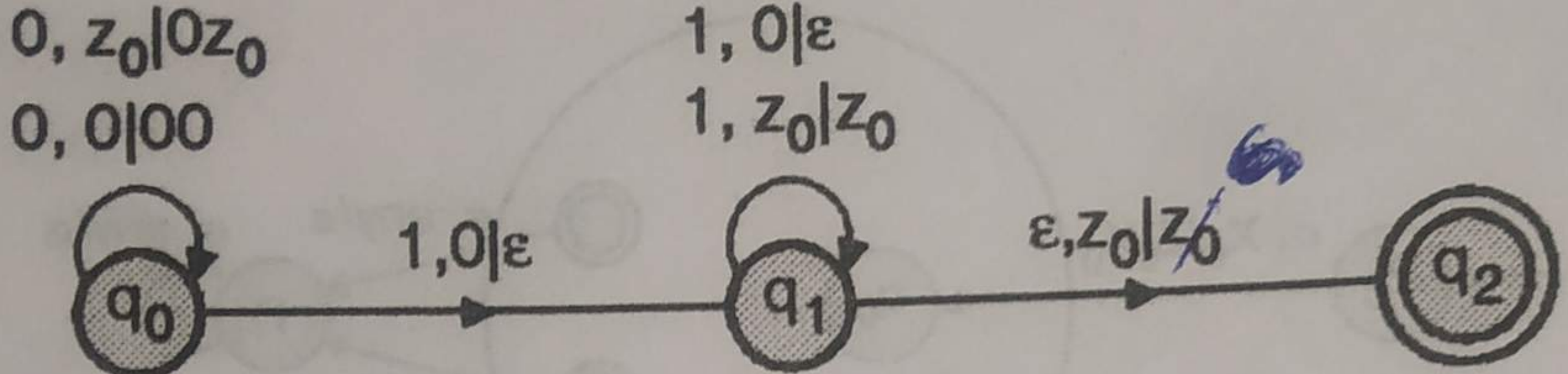

Fig. Ex. 5.4.1(b) : Transition diagram for acceptance through a final state

1. $\delta(q_0, 0, z_0) = (q_0, 0z_0)$
2. $\delta(q_0, 0, 0) = (q_0, 00)$
3. $\delta(q_0, 1, 0) = (q_1, \varepsilon)$
4. $\delta(q_1, 1, 0) = (q_1, \varepsilon)$
5. $\delta(q_1, 1, z_0) = (q_1, z_0)$
6. $\delta(q_1, \varepsilon, z_0) = (q_1, \varepsilon)$

Fig. Ex. 5.4.1(c) : State transition rules for acceptance through an empty stack

1. $\delta(q_0, 0, z_0) = (q_0, 0z_0)$
2. $\delta(q_0, 0, 0) = (q_0, 00)$
3. $\delta(q_0, 1, 0) = (q_1, \varepsilon)$
4. $\delta(q_1, 1, 0) = (q_1, \varepsilon)$
5. $\delta(q_1, 1, z_0) = (q_1, z_0)$
6. $\delta(q_1, \varepsilon, z_0) = (q_2, z_0)$

Fig. Ex. 5.4.1(d) : State transition rules for acceptance through a final state

The PDA accepting through an empty stack is given by :

$M = (\{q_0, q_1\}, \{0, 1\}, \{0, 1, z_0\}, \delta, q_0, z_0, \phi)$

where δ is given in Fig. Ex. 5.4.1(c).

The PDA accepting through final state is given by :

$M = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, z_0\}, \delta, q_0, z_0, \{q_2\})$

where δ is given in Fig. Ex. 5.4.1(d).

Example : Processing of string 00111 by the PDA

Case I : Acceptance through empty stack.

$$(q_0, 00111, z_0) \xrightarrow{\text{(Rule 1)}} (q_0, 0111, 0z_0)$$
$$\xrightarrow{\text{(Rule 2)}} (q_0, 111, 00z_0)$$
$$\xrightarrow{\text{(Rule 3)}} (q_1, 11, 0z_0)$$
$$\xrightarrow{\text{(Rule 4)}} (q_1, 1, z_0)$$
$$\xrightarrow{\text{(Rule 5)}} (q_1, \varepsilon, z_0)$$
$$\xrightarrow{\text{(Rule 6)}} (q_1, \varepsilon, \varepsilon)$$

Case II : Acceptance through final state :

$$(q_0, 00111, z_0) \xrightarrow{\text{(Rule 1)}} (q_0, 0111, 0z_0)$$
$$\xrightarrow{\text{(Rule 2)}} (q_0, 111, 00z_0)$$
$$\xrightarrow{\text{(Rule 3)}} (q_1, 11, 0z_0)$$
$$\xrightarrow{\text{(Rule 4)}} (q_1, 1, z_0)$$
$$\xrightarrow{\text{(Rule 5)}} (q_1, \varepsilon, z_0)$$
$$\xrightarrow{\text{(Rule 6)}} (q_2, \varepsilon, z_0)$$

Fig. Ex. 5.4.1(b) : Transition diagram for acceptance through a final state

1. $\delta(q_0, 0, z_0) = (q_0, 0z_0)$
2. $\delta(q_0, 0, 0) = (q_0, 00)$
3. $\delta(q_0, 1, 0) = (q_1, \varepsilon)$
4. $\delta(q_1, 1, 0) = (q_1, \varepsilon)$
5. $\delta(q_1, 1, z_0) = (q_1, z_0)$
6. $\delta(q_1, \varepsilon, z_0) = (q_1, \varepsilon)$

Fig. Ex. 5.4.1(c) : State transition rules for acceptance through an empty stack

1. $\delta(q_0, 0, z_0) = (q_0, 0z_0)$
2. $\delta(q_0, 0, 0) = (q_0, 00)$
3. $\delta(q_0, 1, 0) = (q_1, \varepsilon)$
4. $\delta(q_1, 1, 0) = (q_1, \varepsilon)$
5. $\delta(q_1, 1, z_0) = (q_1, z_0)$
6. $\delta(q_1, \varepsilon, z_0) = (q_2, z_0)$

Fig. Ex. 5.4.1(d) : State transition rules for acceptance through a final state

Example 5.4.11 SPPU - May 15, 10 Marks

Construct transition table for PDA that accepts the language $L = \{a^{2n} b^n \mid n >= 1\}$. Trace your PDA for the input with $n = 3$.

Solution :

1. For every pair of as z, one x is pushed onto the stack.
2. For every b, one x is popped out from the stack.
3. Final the stack should contain the initial stack symbol z_0.

Transition table

1. $\delta (q_0, a, z_0) = (q_1, z_0)$
2. $\delta (q_1, a, z_0) = (q_0, x z_0)$
3. $\delta (q_0, a, x) = (q_1, x)$
4. $\delta (q_1, a, x) = (q_0, x x)$
5. $\delta (q_0, b, x) = (q_2, \in)$
6. $\delta (q_2, b, x) = (q_2, \in)$
7. $\delta (q_2, \in, z_0) = (q_2, \in)$

accepting through empty stack.

3. $\delta(q_0, a, x) = (q_1, x)$
4. $\delta(q_1, a, x) = (q_0, x\,x)$
5. $\delta(q_0, b, x) = (q_2, \in)$
6. $\delta(q_2, b, x) = (q_2, \in)$
7. $\delta(q_2, \in, z_0) = (q_2, \in)$

accepting through empty stack.

Tracing PDA for a^6b^3

$\delta(q_0, aaaaaabbb, z_0) \overset{\text{Rule 1}}{\Rightarrow} (q_1, aaaaabbb, z_0)$

$\overset{\text{Rule 2}}{\Rightarrow} (q_0, aaaabbb, xz_0)$

$\overset{\text{Rule 3}}{\Rightarrow} (q_1, aaabbb, xz_0)$

$\overset{\text{Rule 4}}{\Rightarrow} (q_0, aabbb, xxz_0)$

$\overset{\text{Rule 3}}{\Rightarrow} (q_1, abbb, xxz_0)$

$\overset{\text{Rule 4}}{\Rightarrow} (q_0, bbb, xxxz_0)$

$\overset{\text{Rule 5}}{\Rightarrow} (q_2, bb, xxz_0)$

$\overset{\text{Rule 6}}{\Rightarrow} (q_2, b, xz_0)$

Example 5.4.11 **SPPU - May 15, 10 Marks**

Construct transition table for PDA that accepts the language $L = \{a^{2n} b^n \mid n >= 1\}$. Trace your PDA for the input with n = 3.

Solution :

1. For every pair of as z, one x is pushed onto the stack.
2. For every b, one x is popped out from the stack.
3. Final the stack should contain the initial stack symbol z_0.

Transition table

1. $\delta(q_0, a, z_0) = (q_1, z_0)$
2. $\delta(q_1, a, z_0) = (q_0, x\,z_0)$
3. $\delta(q_0, a, x) = (q_1, x)$
4. $\delta(q_1, a, x) = (q_0, x\,x)$
5. $\delta(q_0, b, x) = (q_2, \in)$
6. $\delta(q_2, b, x) = (q_2, \in)$
7. $\delta(q_2, \in, z_0) = (q_2, \in)$

accepting through empty stack.

Rule 6

$\Rightarrow (q_2, \in, z_0)$

Rule 7

$\Rightarrow (q_2, \in, \in)$

Example 5.4.5 SPPU - May 12, 8 Marks

Let $L = \{a^n b^n c^m d^m \mid n, m \geq 1\}$ find a PDA that accepts L.

Solution : Algorithm

1. Sequence of a's should be pushed onto the stack.
2. For every b as input, an 'a' should be erased from the stack.
3. Sequence of c's should be pushed onto the stack.
4. For every d as input, a 'c' should be erased from the stack.

The PDA is given by :

$$M = (\{q_0, q_1, q_2, q_3\}, \{a, b, c, d\}, \{a, c, z_0\}, \delta, q_0, z_0, \phi)$$

Where the transition function δ is given below :

$\delta(q_0, a, z_0) = (q_0, az_0)$ [Push the first a]

$\delta(q_0, a, a) = (q_0, aa)$ [Push remaining a's]

$\delta(q_0, b, a) = (q_1, \varepsilon)$ [Erase an 'a' on first b]

$\delta(q_1, b, a) = (q_1, \varepsilon)$

[Erase remaining a's on subsequent b's]

$\delta(q_1, c, z_0) = (q_2, cz_0)$ [First c is pushed]

$\delta(q_2, c, c) = (q_2, cc)$ [Subsequent c's are pushed]

$\delta(q_2, d, c) = (q_3, \varepsilon)$

[On first d, machine transits to q_3 with a pop]

$\delta(q_3, d, c) = (q_3, \varepsilon)$ [For every d, a 'c' is erased]

$\delta(q_3, \varepsilon, z_0) = (q_3, \varepsilon)$

[String is accepted through empty stack]