SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 OF THE UGC ACT, 1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

CSE211 – Formal Languages and Automata Theory

U2L6_Constuction of Push Down Automata (PDA)

Dr. P. Saravanan

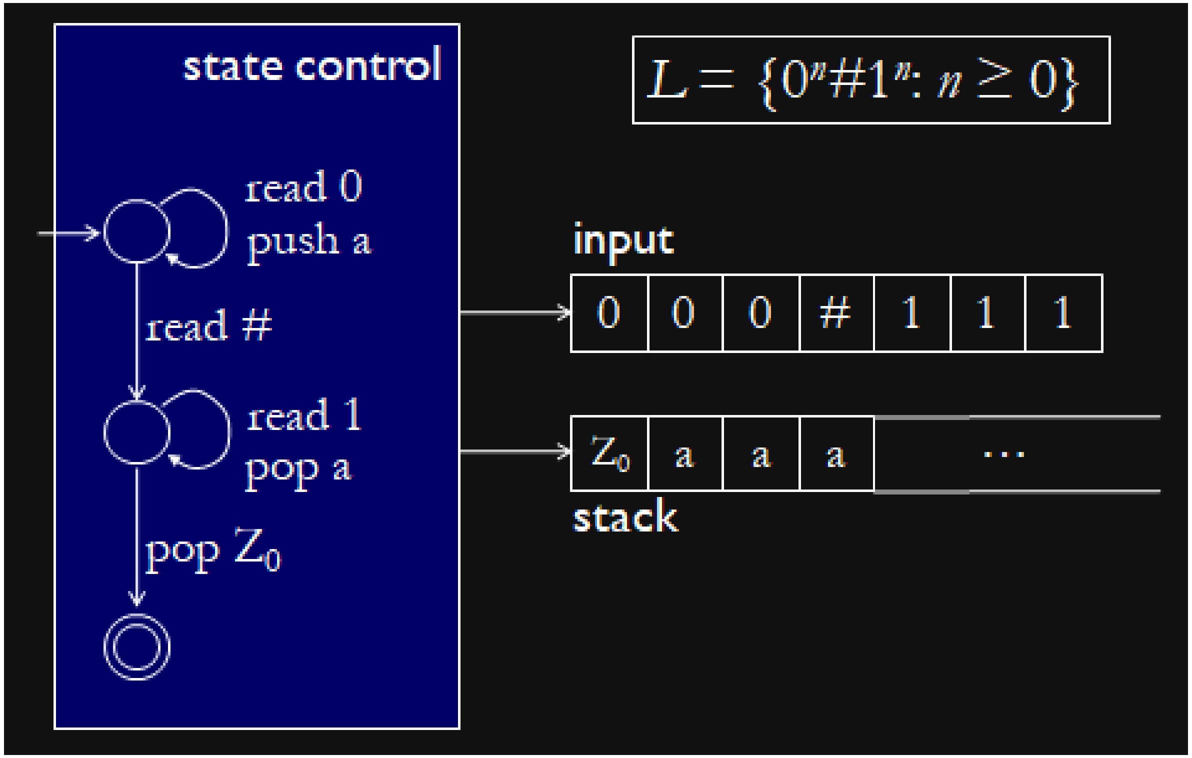
School of Computing

SASTRA Deemed University

Agenda

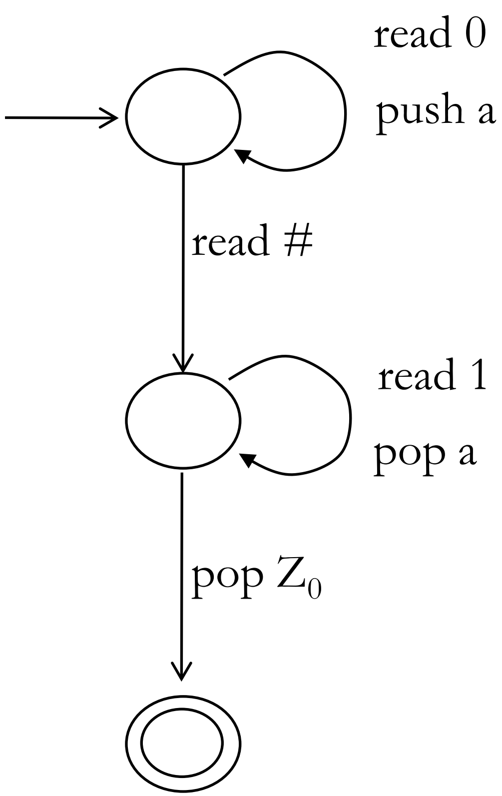
- Recap of previous class
- Introduction
- Definition of PDA
- The Language of a PDA
- Equivalence of PDA’s and CFG’s
- Deterministic PDA’s

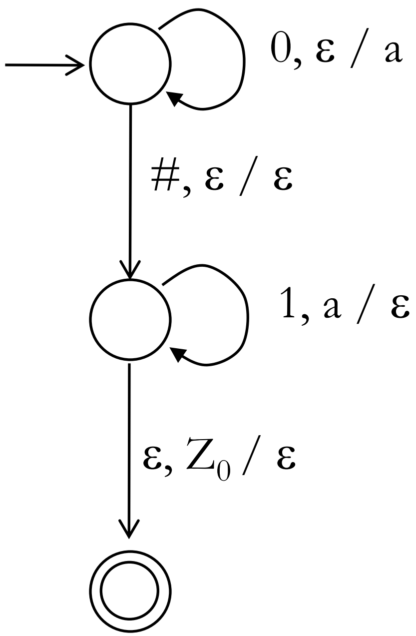
Example



state control
read 0
push a
read #
read 1
pop a
pop Z_0
$L = \{0^n\#1^n : n \geq 0\}$
input
0 0 0 # 1 1 1
Z_0 a a a ...
stack

Shorthand notation

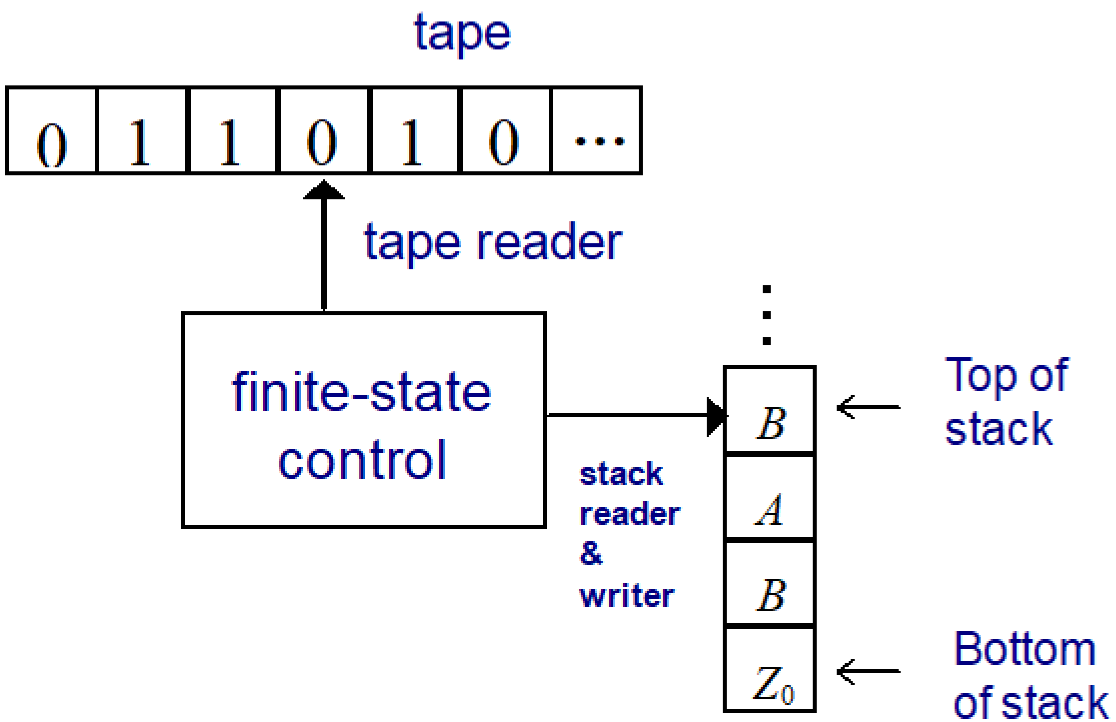




read, pop / push

Graphical Model of PDA

- A graphic model of a PDA



A graph model of a PDA

Formal Definition

A PDA is a 7-tuple $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ where

- Q: a finite set of states
- Σ: a finite set of input symbols
- Γ: a finite stack alphabet
- δ: a transition function such that $\delta(q, a, X)$ **is a set of pairs** (p, γ) where
  - $q \in Q$ (the current state)
  - $a \in \Sigma$ or $a = \varepsilon$ (an input symbol or an empty string)
  - $X \in \Gamma$
  - $p \in Q$ (the next state)

Formal Definition…

- $\gamma \in \Gamma^*$ which replaces X on the top of the stack:

  when $\gamma = \varepsilon$, the top stack symbol is popped up

  when $\gamma = X$, the stack is unchanged

  when $\gamma = YZ$, X is replaced by Z, and Y is pushed to the top

  when $\gamma = \alpha Z$, X is replaced by Z and string α is pushed to the top
- q_0: the start state
- Z_0: the start symbol of the stack
- F: the set of accepting or final states

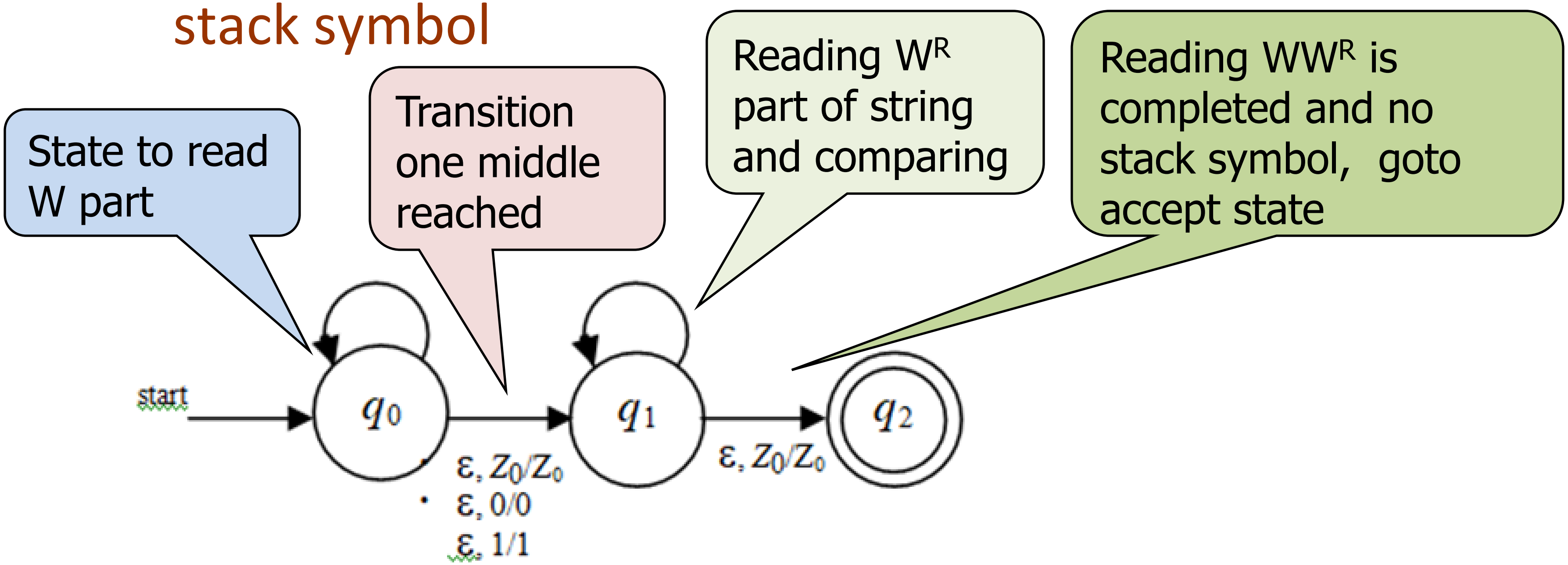
Designing PDA means defining all these elements

Designing PDA: Example

- Example: 6.1 - Design a PDA to accept the language Lwwr = $\{ww^R \mid w \text{ is in } (0 + 1)^*\}$
- In start state q0, copy input symbols onto the stack
- At any time, nondeterministically guess whether the middle of ww^R is reached and enter q1, or continue copying input symbols.
- In q1, compare remaining input symbols with those on the stack one by one.
- If the stack can be so emptied, then the matching of w with w^R succeeds.

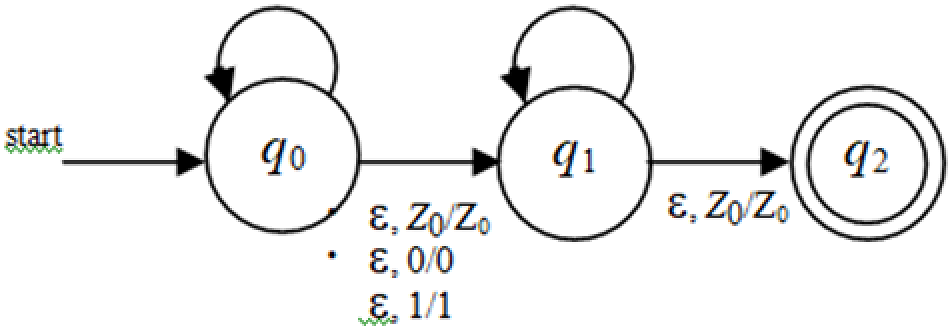
Designing PDA Example

- Designing a PDA to accept the language $L_{ww}{}^R$. Where $\Sigma=\{0,1\}$ and $\Gamma=\{a,b\}$
  - With stack symbol – use a for 0 and use b for 1
  - Without stack symbol – use 0 for 0 and use 1 for 1 as stack symbol



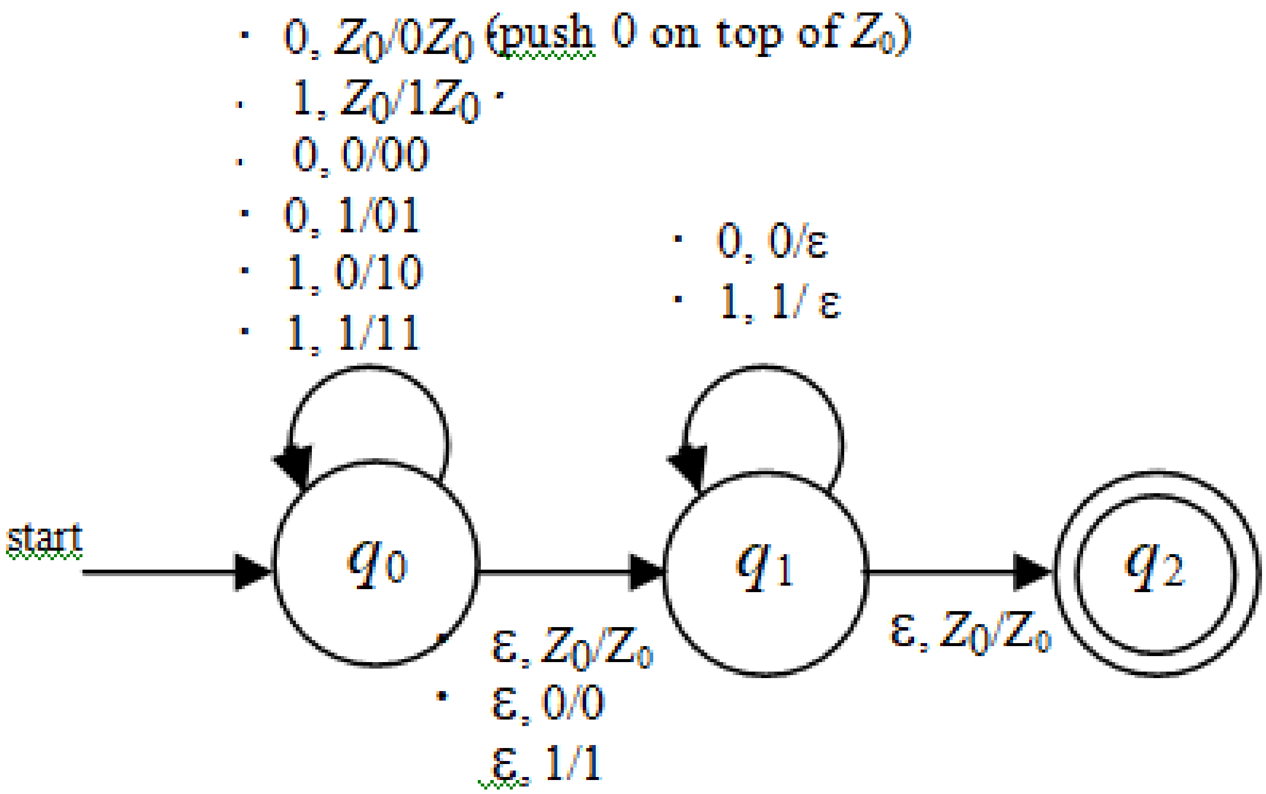
Designing PDA Example

- 10100101



Designing PDA Example

- Designing a PDA to accept the language L_{ww^R}.



Designing PDA: Example

- Designing a PDA to accept the language $L_{ww}{}^R$.
  - Need a start symbol Z of the stack and a 3rd state q_2 as the accepting state.
  - $P = (\{q_0, q_1, q_2\}, \{0, 1\}, \{0, 1, Z_0\}, \delta, q_0, Z_0, \{q_2\})$ such that
    - $\delta(q_0, 0, Z_0) = \{(q_0, 0Z_0)\}$, $\delta(q_0, 1, Z_0) = \{(q_0, 1Z_0)\}$

      (initial pushing steps with Z_0 to mark stack bottom)
    - $\delta(q_0, 0, 0) = \{(q_0, 00)\}$, $\delta(q_0, 0, 1) = \{(q_0, 01)\}$,

      $\delta(q_0, 1, 0) = \{(q_0, 10)\}$, $\delta(q_0, 1, 1) = \{(q_0, 11)\}$

Rules for pushdown automata

- $\delta(q_0, \varepsilon, Z_0) = \{(q_1, Z_0)\}$

  (check if input is ε which is in L_{ww^R})

- $\delta(q_0, \varepsilon, 0) = \{(q_1, 0)\}$, $\delta(q_0, \varepsilon, 1) = \{(q_1, 1)\}$

  (check the string's middle)

- $\delta(q_1, 0, 0) = \{(q_1, \varepsilon)\}$, $\delta(q_1, 1, 1) = \{(q_1, \varepsilon)\}$

  (matching pairs)

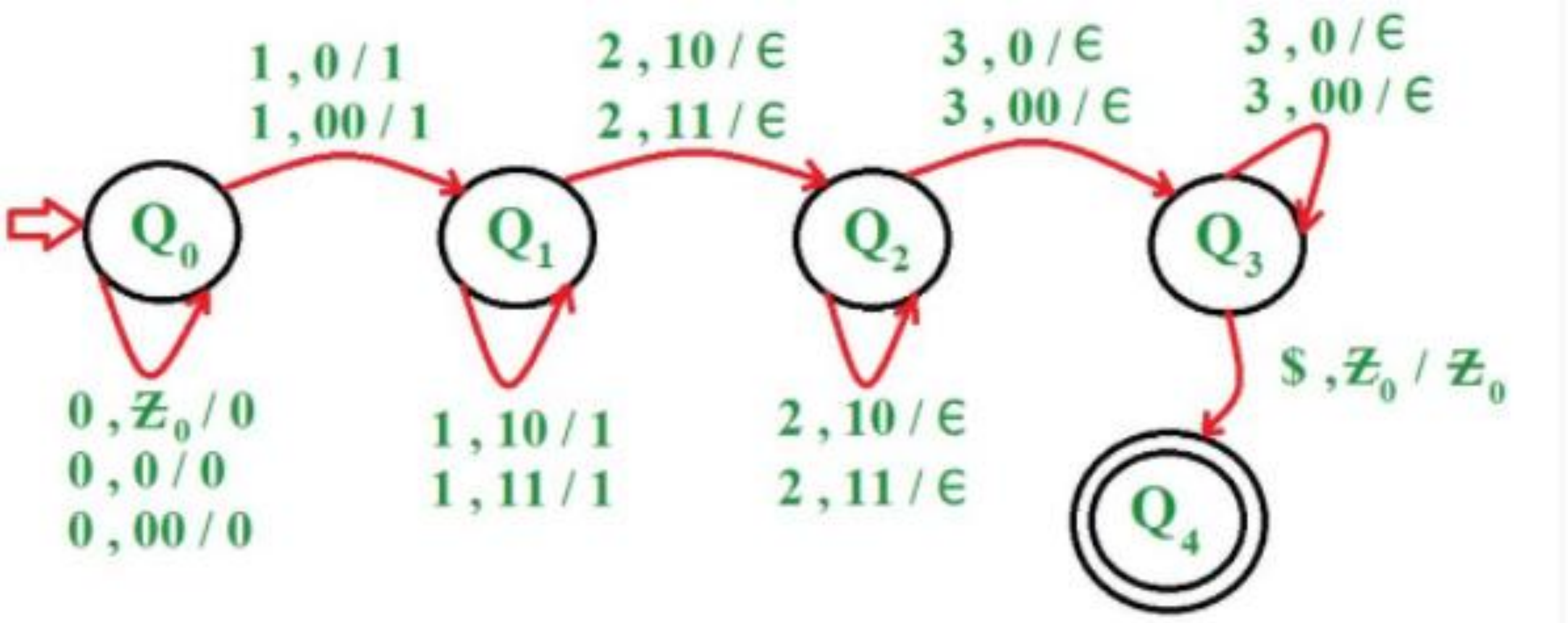
- $\delta(q_1, \varepsilon, Z_0) = \{(q_2, Z_0)\}$

  (entering final state)

Example 2

- Construct a PDA for language $L = \{0^n 1^m 2^m 3^n \mid n>=1, m>=1\}$
- Steps:
  - Step-1: On receiving 0 push it onto stack. On receiving 1, push it onto stack and goto next state
  - Step-2: On receiving 1 push it onto stack. On receiving 2, pop 1 from stack and goto next state
  - Step-3: On receiving 2 pop 1 from stack. If all the 1's have been popped out of stack and now receive 3 then pop a 0 from stack and goto next state
  - Step-4: On receiving 3 pop 0 from stack. If input is finished and stack is empty then goto last state and string is accepted

Example 2



1 , 0 / 1
1 , 00 / 1
2 , 10 / ∈
2 , 11 / ∈
3 , 0 / ∈
3 , 00 / ∈
3 , 0 / ∈
3 , 00 / ∈
Q_0
Q_1
Q_2
Q_3
Q_4
0 , Z_0 / 0
0 , 0 / 0
0 , 00 / 0
1 , 10 / 1
1 , 11 / 1
2 , 10 / ∈
2 , 11 / ∈
$, Z_0 / Z_0

Example 3

- Design a PDA for accepting a language $\{a^n b^{2n} \mid n>=1\}$.
- In this language, n number of a's should be followed by 2n number of b's.
- Hence, we will apply a very simple logic, and that is if we read single 'a', we will push two a's onto the stack.
- As soon as we read 'b' then for every single 'b' only one 'a' should get popped from the stack.

Example 3

$\delta(q0, a, Z) = (q0, aaZ)$

$\delta(q0, a, a) = (q0, aaa)$

Now when we read b, we will change the state from q0 to q1 and start popping corresponding 'a'. Hence,

$\delta(q0, b, a) = (q1, \varepsilon)$

Thus this process of popping 'b' will be repeated unless all the symbols are read. Note that popping action occurs in state q1 only.

$\delta(q1, b, a) = (q1, \varepsilon)$

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:

Example 3

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:

$\delta(q1, \varepsilon, Z) = (q2, \varepsilon)$

Where

PDA = $(\{q0, q1, q2\}, \{a, b\}, \{a, Z\}, \delta, q0, Z, \{q2\})$

$\delta(q0, a, Z) = (q0, aaZ)$

$\delta(q0, a, a) = (q0, aaa)$

$\delta(q0, b, a) = (q1, \varepsilon)$

$\delta(q1, b, a) = (q1, \varepsilon)$

$\delta(q1, \varepsilon, Z) = (q2, \varepsilon)$

Example 3

- simulate this PDA for the input string "aaabbbbbb".

$\delta(q0, aaabbbbbb, Z) \vdash \delta(q0, aabbbbbb, aaZ)$

$\vdash \delta(q0, abbbbbb, aaaaZ)$

$\vdash \delta(q0, bbbbbb, aaaaaaZ)$

$\vdash \delta(q1, bbbbb, aaaaaZ)$

$\vdash \delta(q1, bbbb, aaaaZ)$

$\vdash \delta(q1, bbb, aaaZ)$

$\vdash \delta(q1, bb, aaZ)$

$\vdash \delta(q1, b, aZ)$

$\vdash \delta(q1, \varepsilon, Z)$

$\vdash \delta(q2, \varepsilon)$

ACCEPT

Summary

- Definition of PDA
- Designing of PDA

References

- John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Pearson, 3rd Edition, 2011.
- Peter Linz, An Introduction to Formal Languages and Automata, Jones and Bartle Learning International, United Kingdom, 6th Edition, 2016.

Next Class:

ID, Language, Equivalence

THANK YOU.