BCSE304L - Theory of Computation

Dr. R. Arumuga Arun,

Cabin : PRP 315(A&B)-19,

Mailid : arumugaarun.r@vit.ac.in.

Module 5 –
Pushdown Automata

Definition of the Pushdown automata - Languages of a Pushdown automata – Power of Non-Deterministic Pushdown Automata and Deterministic pushdown automata

Topic: Non-Deterministic Pushdown Automata and Deterministic pushdown automata

Can be possible with DPDA

$L = \{ wcw^R \,/\, w \in \{0, 1\}^* \}$

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

$Q = \{q_0, q_1\}$, $\Sigma = \{0, 1, c\}$, $\Gamma = \{X, Y, Z_0\}$

Assume, $w = 001$, $w^R = 100$ $\quad wcw^R = 001c100$

Can be possible with DPDA

$L = \{ wcw^R / w \in \{ 0, 1\}^* \}$

$M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, \phi)$, where $Q = \{q_0, q_1\}$, $\Sigma = \{0, 1, c\}$, $\Gamma = \{X, Y, Z_0\}$

Assume, $w = 001$, $w^R = 100$ $wcw^R = 001c100$

$\delta(q_0, 0, Z_0) = \{ (q_0, X Z_0) \}$

$\delta(q_0, 0, X) = \{ (q_0, X X) \}$

$\delta(q_0, 0, Y) = \{ (q_0, X Y) \}$

$\delta(q_0, c, X) = \{ (q_1, X) \}$

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

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

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

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

$\delta(q_0, 1, Z_0) = \{ (q_0, Y Z_0) \}$

$\delta(q_0, 1, Y) = \{ (q_0, YY) \}$

$\delta(q_0, 1, X) = \{ (q_0, YX) \}$

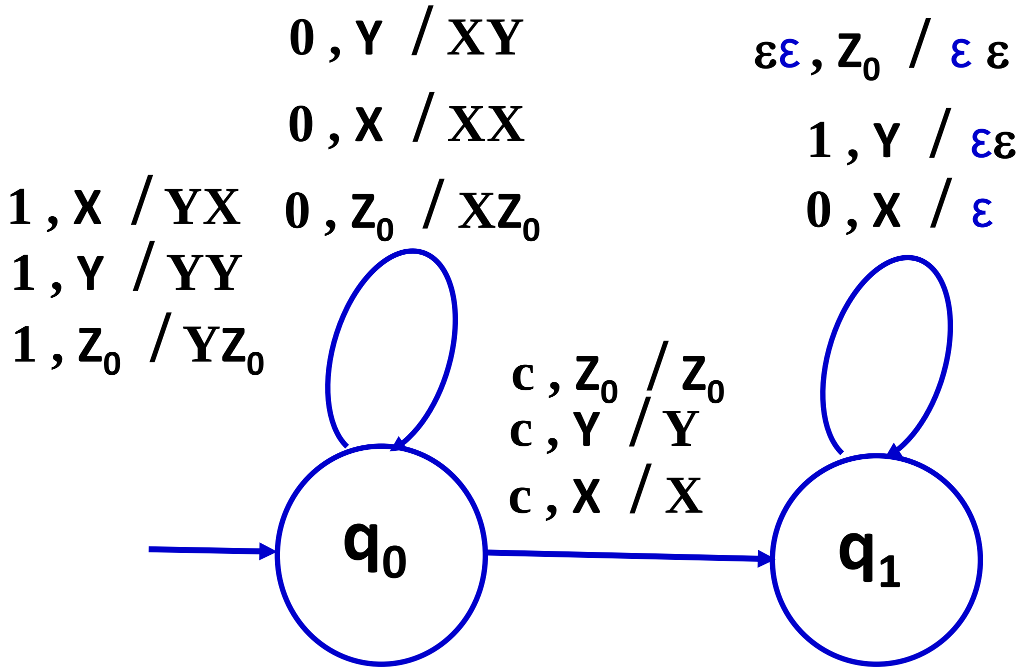
$\delta(q_0, c, Y) = \{ (q_1, Y) \}$

$w = 001$, $w^R = 100$ $wcw^R = 001c100$

$$(q_0 , 001c100 , Z_0)$$
$$\vdash (q_0 , 01c100 , XZ_0)$$
$$\vdash (q_0 , 1c100 , XXZ_0)$$
$$\vdash (q_0 , c100 , YXXZ_0)$$
$$\vdash (q_1 , 100 , YXXZ_0)$$
$$\vdash (q_1 , 00 , XXZ_0)$$
$$\vdash (q_1 , 0 , XZ_0)$$
$$\vdash (q_1 , \varepsilon , Z_0)$$
$$\vdash (q_1 , \varepsilon , \varepsilon)$$

$\delta(q_0 , 0 , Z_0) = \{ (q_0 , X Z_0) \}$
$\delta(q_0 , 0 , X) = \{ (q_0 , X X) \}$
$\delta(q_0 , 0 , Y) = \{ (q_0 , X Y) \}$
$\delta(q_0 , c , X) = \{ (q_1 , X) \}$
$\delta(q_0 , c , Z_0) = \{ (q_1 , Z_0) \}$
$\delta(q_1 , 0 , X) = \{ (q_1 , \varepsilon) \}$
$\delta(q_1 , 1 , Y) = \{ (q_1 , \varepsilon) \}$
$\delta(q_1 , \varepsilon, Z_0) = \{ (q_1 , \varepsilon) \}$

$\delta(q_0 , 1 , Z_0) = \{ (q_0 , Y Z_0) \}$
$\delta(q_0 , 1 , Y) = \{ (q_0 , YY) \}$
$\delta(q_0 , 1 , X) = \{ (q_0 , YX) \}$
$\delta(q_0 , c , Y) = \{ (q_1 , Y) \}$



$\delta(q_0 , 0 , Z_0) = \{ (q_0 , X Z_0) \}$

$\delta(q_0 , 0 , X) = \{ (q_0 , X X) \}$

$\delta(q_0 , 0 , Y) = \{ (q_0 , X Y) \}$

$\delta(q_0 , c , X) = \{ (q_1 , X) \}$

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

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

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

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

$\delta(q_0 , 1 , Z_0) = \{ (q_0 , Y Z_0) \}$

$\delta(q_0 , 1 , Y) = \{ (q_0 , YY) \}$

$\delta(q_0 , 1 , X) = \{ (q_0 , YX) \}$

$\delta(q_0 , c , Y) = \{ (q_1 , Y) \}$

Whether it can be possible with DPDA ?

$L = \{ ww^R \,/\, w \in \{0, 1\}^* \}$

Assume, i) $w = aa$, $w^R = aa$, $ww^R = aaaa$

ii) $w = abb$, $w^R = bba$, $ww^R = abbbba$

Whether it can be possible with DPDA ?

$L = \{ ww^R / w \in \{0, 1\}^* \}$

Assume, i) w = aa, w^R = aa, ww^R = aaaa

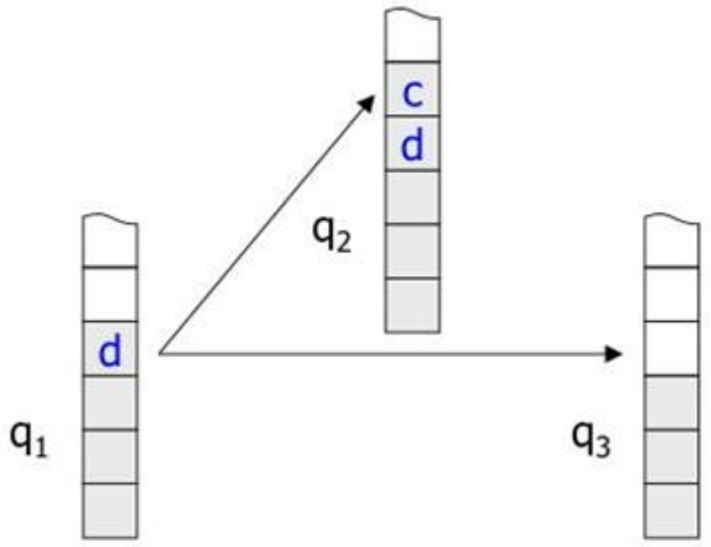
ii) w = abb, w^R = bba, ww^R = abbbba

Not possible since we can't find the middle of the word

Non-Deterministic PDA

- More than one move from a state on an input symbol and stack symbol.
- A **non-deterministic** PDA is used to generate a language that a deterministic automata cannot generate.
  - It is more powerful than a deterministic PDA
- Example

$$\delta(q_1, a, d) = \{(q_2, cd), (q_3, \lambda)\}$$



Representation

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

Q: finite set of internal states

Σ: finite set of symbols - input alphabet

Γ: finite set of symbols - stack alphabet

$\delta: Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \rightarrow$ finite subsets of $Q \times \Gamma^*$ transition function

$q_0 \in Q$: initial state

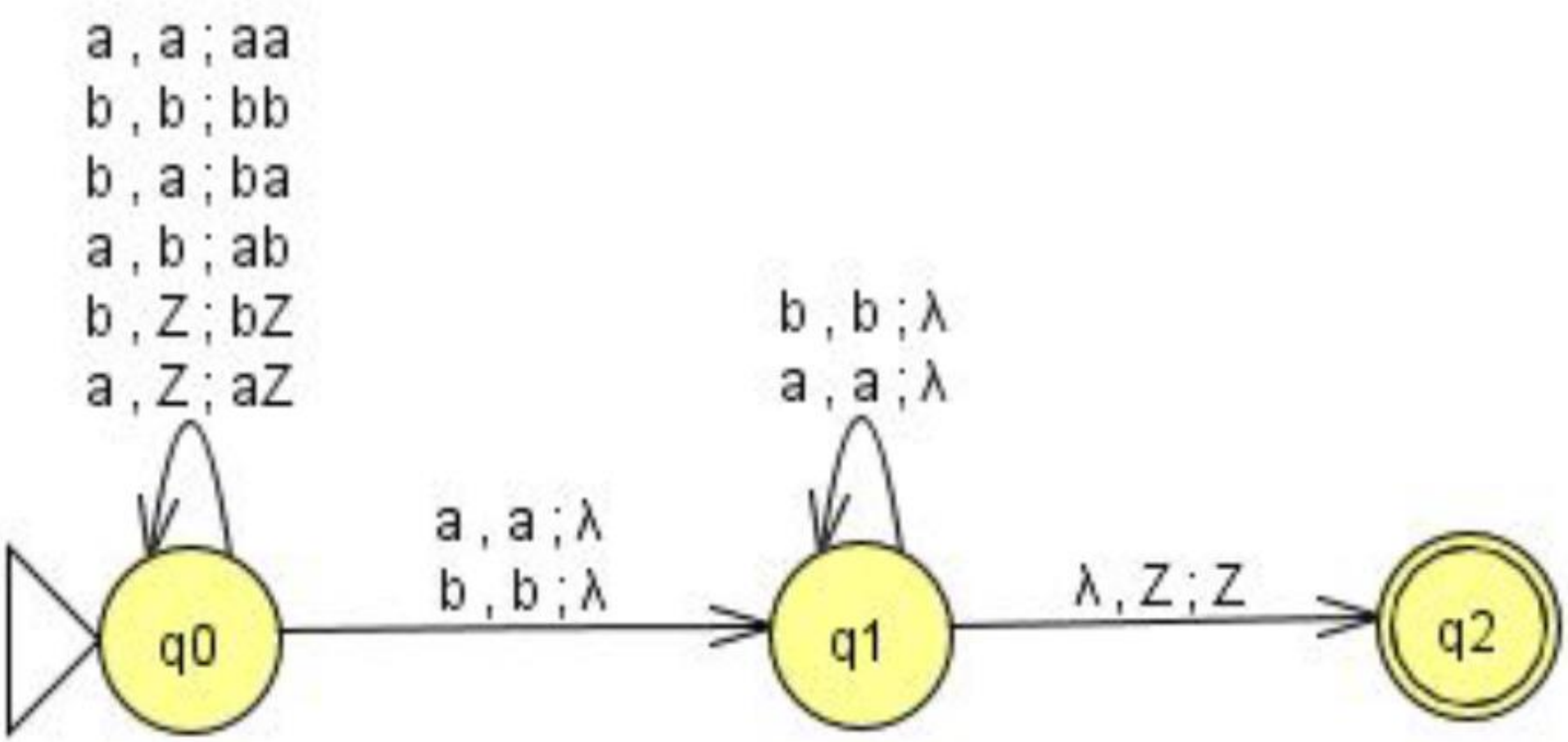
$z \in \Gamma$: stack start symbol

$F \subseteq Q$: set of final states

$$\delta: Q \times (\Sigma \cup \{\lambda\}) \times \underbrace{\Gamma}_{\text{stack top}} \rightarrow \text{finite subsets of } Q \times \underbrace{\Gamma^*}_{\text{stack top replacement}}$$

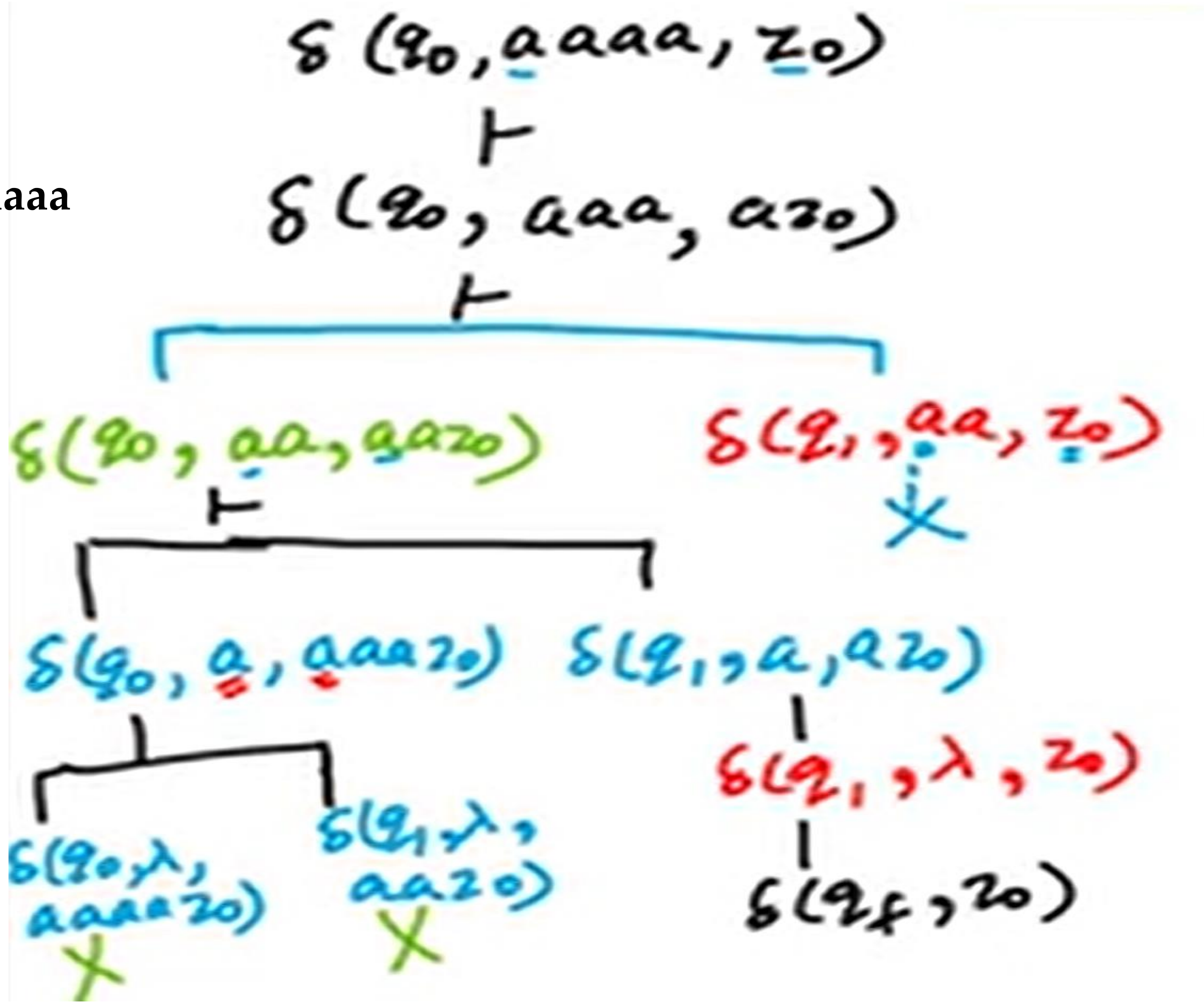
Example 1

- Construct a NPDA for a language, L = {WW^R | w = $\{0,1\}^+$}



Cont…

Check the acceptance of the string, **w = aaaa**



Check the acceptance for the string abbbba

$\delta(q_0, abbbba, z_0)$ ✓

$\delta(q_0, \cancel{a}bbbba, az_0)$ ✓

$\delta(q_0, \cancel{b}bbba, baz_0)$ ✓ $\delta(q_1, bbbba, z_0)$ ✗

$\delta(q_0, \cancel{b}bba, bbaz_0)$ ✓ $\delta(q_1, \cancel{b}bba, az_0)$

$\delta(q_0, \cancel{b}ba, bbbaz_0)$ $\delta(q_1, \cancel{b}ba, baz_0)$ ✓ $\delta(q_0, \cancel{b}bba, baz_0)$ $\delta(q_1, \cancel{b}bba, z_0)$ ✗

$\delta(q_0, \cancel{b}a, bbbbaz_0)$ ✗ $\delta(q_1, \cancel{b}a, bbaz_0)$ ✗ $\delta(q_0, \cancel{b}a, bbaz_0$ ✗ $\delta(q_0, \cancel{b}a, az_0)$ ✓

$\delta(q_0, a, az_0)$ $\delta(q_0, bba, baz_0)$

$\delta(q_0, \varepsilon, aaz_0)$ ✗ $\delta(q_0, \varepsilon, z_0)$ ✓ $\delta(q_0, \cancel{b}ba, bbaz_0)$ ✗ $\delta(q_1, \cancel{b}ba, az_0)$ ✗

Cont…

- Check the acceptance of the string, w = abbbba

$$
\begin{aligned}
(q_0, abbbba, z) \quad & \vdash (q_0, bbbba, az) \\
& \vdash (q_0, bbba, baz) \\
& \vdash (q_0, bba, bbaz) \\
& \vdash (q_1, ba, baz) \\
& \vdash (q_1, a, az) \\
& \vdash (q_1, \varepsilon, z) \\
& \vdash (q_2, \varepsilon, z)
\end{aligned}
$$

So, at the end, the stack becomes empty then we can say that the string is accepted by the PDA.

Cont…

- Check the acceptance of the string, w = abbbba

$(q_0, abbbba, z) \vdash (q_0, bbbba, az)$

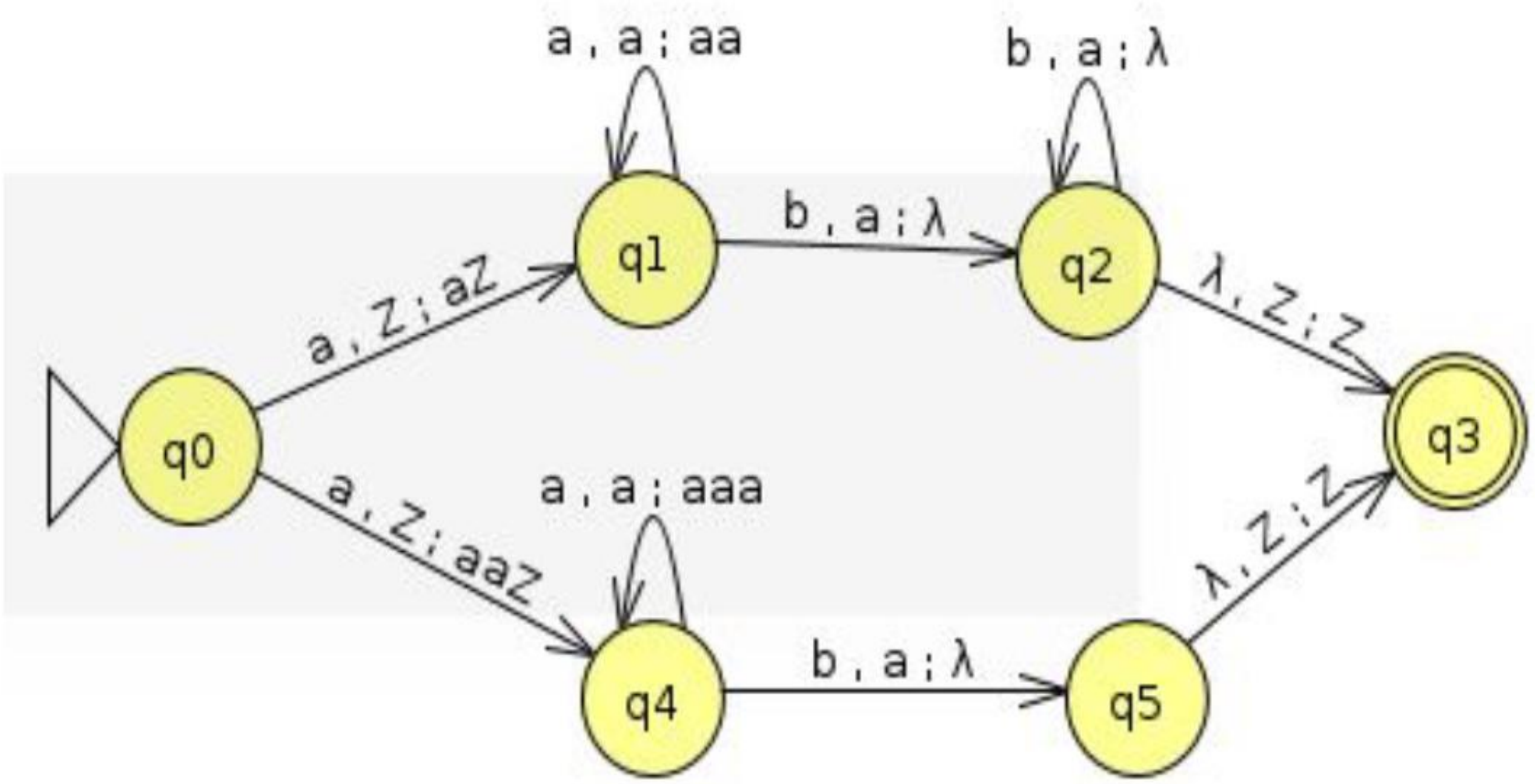
$\vdash (q_0, bbba, baz)$

$\vdash (q_1, bba, az)$

No transition exists [Dead configuration]

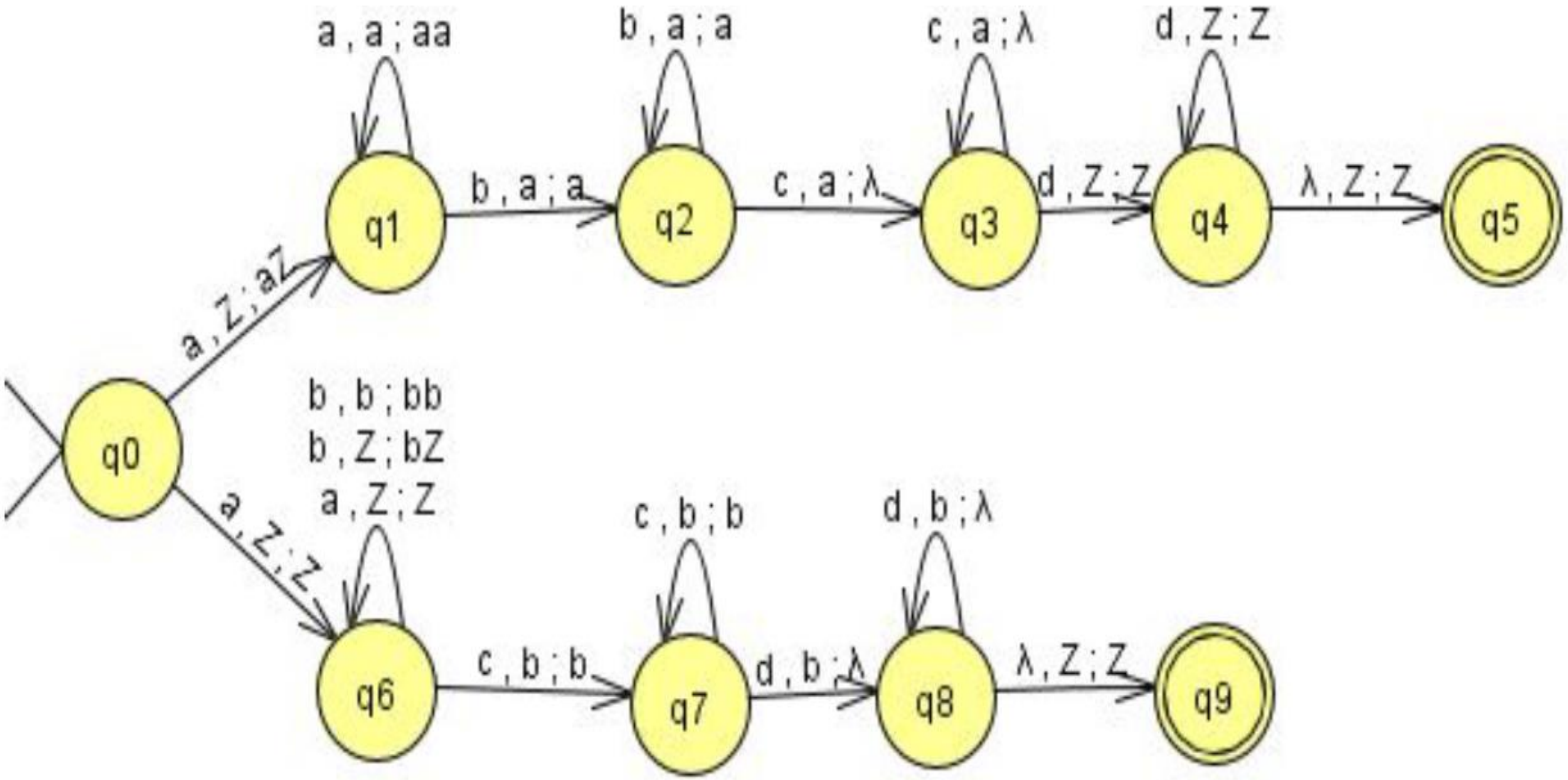
Example 2

- Construct a NPDA for a language, L = $\{a^n b^n \mid n>=1\} \cup \{a^n b^{2n} \mid n>=1\}$



Example 3

- Construct a NPDA for a language, $L = \{a^i b^j c^k d^l \mid i=k \text{ or } j=l, i>=1, j>=1\}$



Try it yourself

1. Design a NPDA for accepting the language L = {a^m b^n c^p d^q | m + n = p + q : m, n, p, q>=1}
2. Design a NPDA for accepting the language L = {a^m b^n c^{(m+n)} | m,n ≥ 1}
3. Design a NPDA for accepting the language L = {a^{m} b^{2m+1} | m ≥ 1}, or,L = {a^{m} b b^{2m} | m ≥ 1}
4. Design a NPDA for accepting the language L = {a^2m b^3m | m ≥ 1}