Move to the final state.

$\delta 4\ (q_{f3}, \varepsilon, x) = (q_e, \varepsilon, \varepsilon)$

$\delta 4\ (q_e, \varepsilon, x) = (q_e, \varepsilon, \varepsilon)$

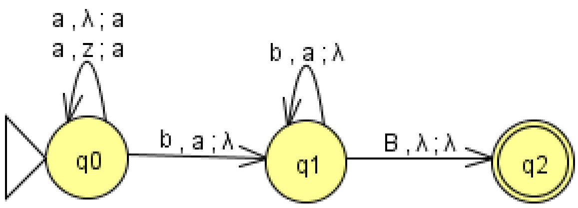
for all $x \in \Gamma$.

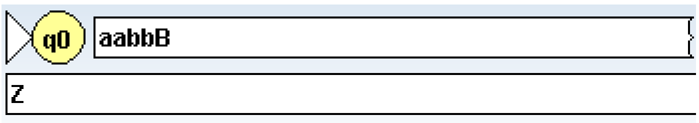
To show $x \in L(M4)$

$(q4, x, z4) \rightarrow (q3, x, z3\ z4) \rightarrow (q_{f3}, \varepsilon, \gamma) \rightarrow (q_e, \varepsilon, \gamma)$

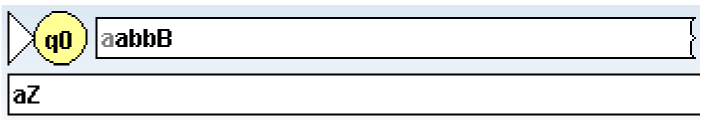
Example 1 Design a PDA which accepts $L = \{a^n b^n / n \geq 1]$

Solution:

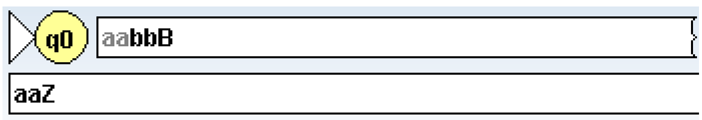




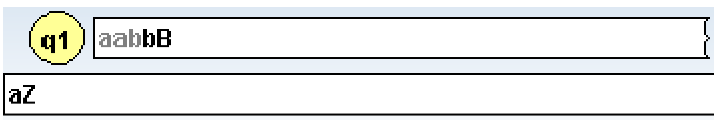
Initial state



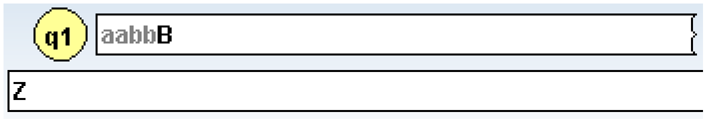
Read a on the input tape and push ‘a’ into the stack, a,z;a



Read a on the input tape and push ‘a’ into the stack, a,λ;a



Read b on the input tape, pop ‘a’ from the stack and state changes to q1, b,a;λ

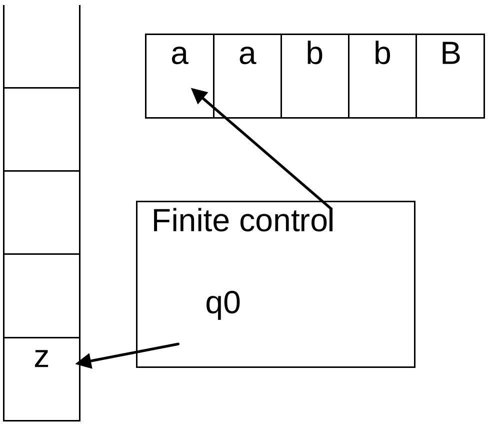


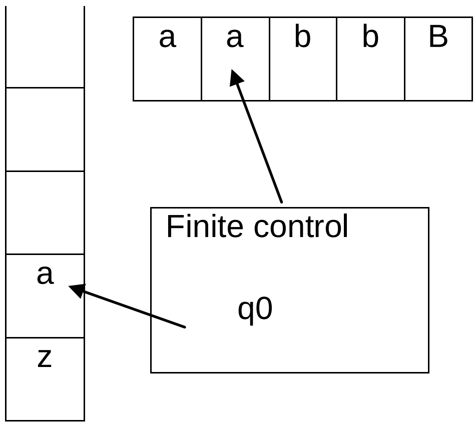
Read b on the input tape, pop ‘a’ from the stack and state remains in q1, b,a;λ

B,λ;λ

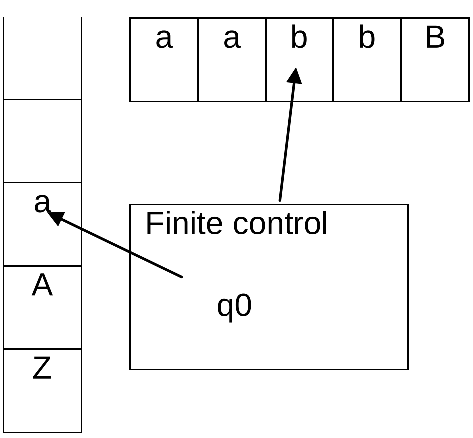
Read B(blank) on the input tape, reaches the final state and state changes to q2.

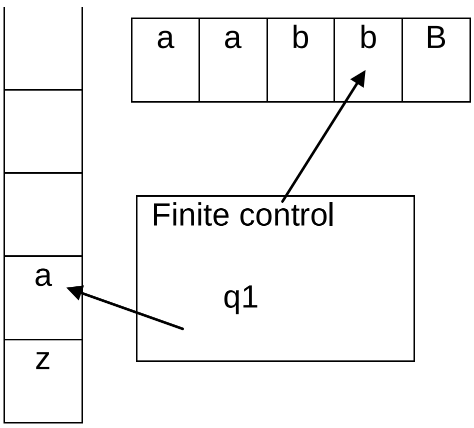
Another method



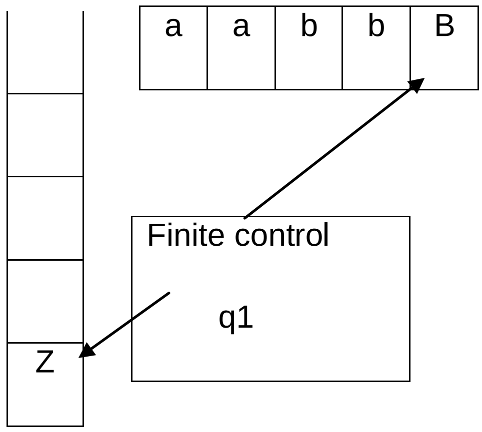


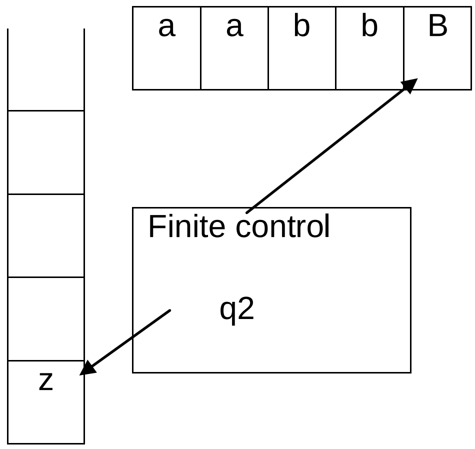
Step 1: Initial configuration Step 2: a is pushed





Step 3: a is pushed Step 4: pop a and changes to state q1





Step 5: pop a

Step 6: Changes to final state and halts

The PDA moves are as follows

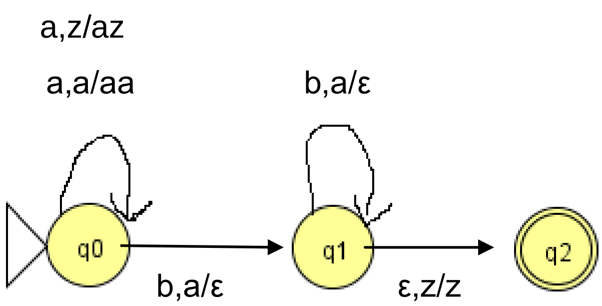
δ(q0,a,z) = (q0,az) push a

δ(q0,a,a) = (q0,aa) push a

δ(q0,b,a) = (q1,ε) pop a and change the state

δ(q1,b,a) = (q1,ε) pop a

δ(q1,ε,z) = (q2,z) change to the final state q2 and halt



We will stimulate this PDA for the following string.

(q0,aabb,z) ├ (q0,abb,az)

├ (q0,bb,aaz)

├ (q1,b,az)

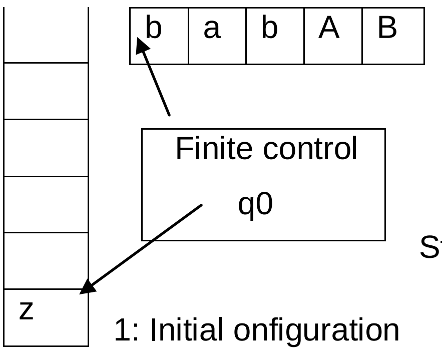
├ (q1,ε,z)

├ (q2,B)

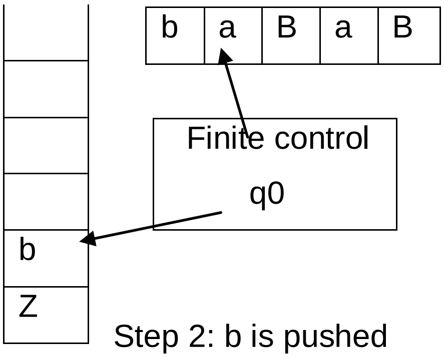
Example 2: Design a PDA which accepts equal number of a's and b's over ∑= {a, b}

Solution

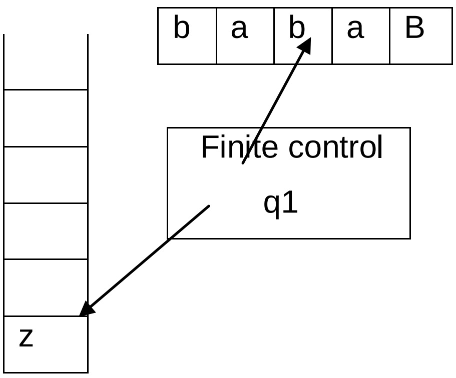
Here input can start either with a or b and they can occur in any order as bbaaba or bababa etc. to design such a PDA, read first symbol either a or b on the tape and push that symbol in to the stack. Then read the next symbol. If the next symbol on the tape is different from the top of the stack then pop the symbol from the top of the stack. When the entire tape is read and reaches the blank symbol, if the stack becomes empty, then it is said to be successful.

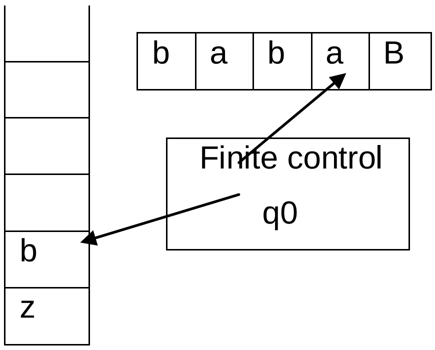


Step 1: Initial onfiguration

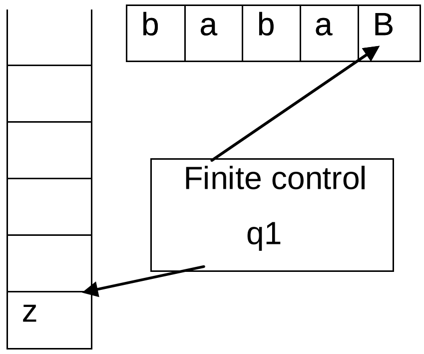


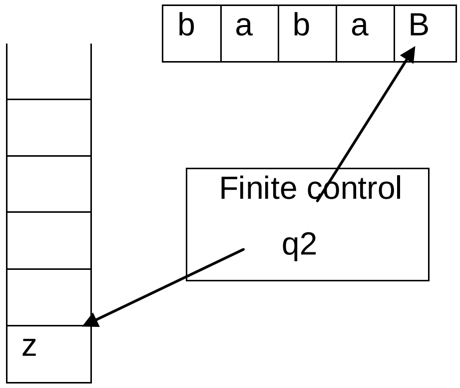
Step 2: b is pushed





Step 3: pop b and states changes to q1 Step 4: push b and state changes to q0





Step 5: pop b and states changes to q1 Step 6: reaches the final state q2 and halts

The PDA moves are as follows:-

Let q0 be the initial state, q2 be the final state and z be the bottom of the stack.

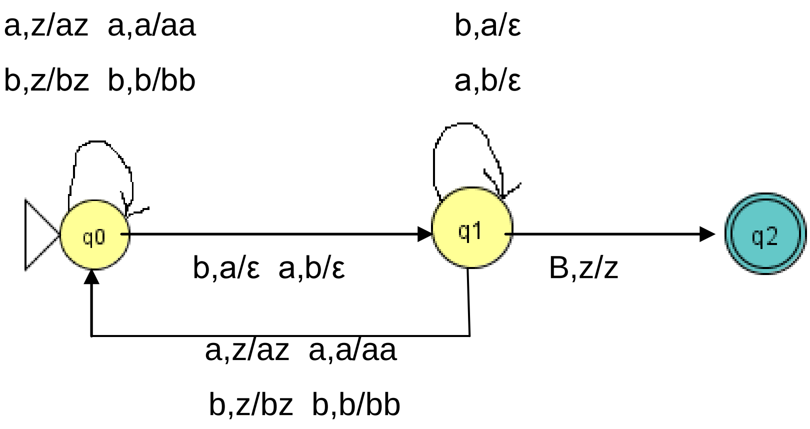
δ(q0.b.z) = (q0,bz)

δ(q0,a,b) = (q1,ε)

δ(q1,b,z) = (q0,bz)

δ(q0,a,b) = (q1,ε)

δ(q1,Bz) = (q2,z)



We will stimulate this for the following string abba

δ (q0,abba,z) ├ (q0,bba,az)

$\vdash$ (q1,ba,z)

$\vdash$ (q0,a,bz)

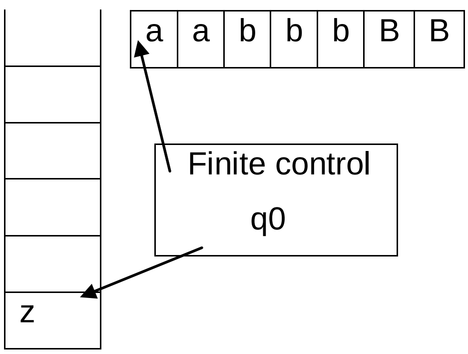
$\vdash$ (q1,ε,z)

$\vdash$ (q2,B)

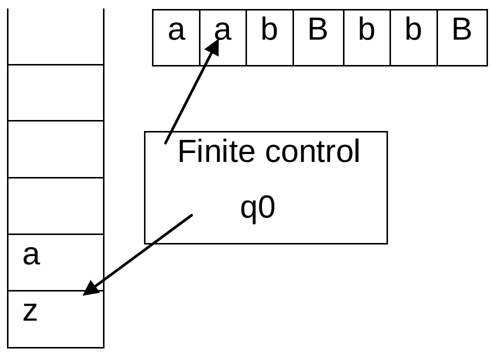
Example 3: Construct PDA for the language L = {$a^n b^n$|n$\geq$1}

Solution:

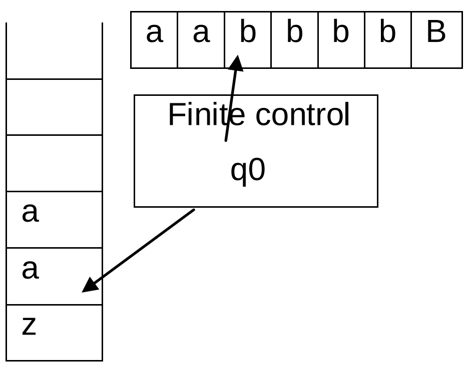
Here we have to match one ‘a’ with two ‘b’s . Now push ‘a’ in to the stack. After that, read one ‘b’ and change to different state. Then read second’ b’, now pop ‘a’ from the stack and reset the state to the initial state, to continue the same process with each ‘b’. when the input read is completed, then if stack becomes empty it is said to be successful.



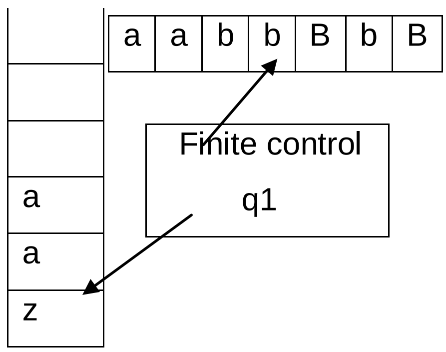
Step 1: Initial configuration



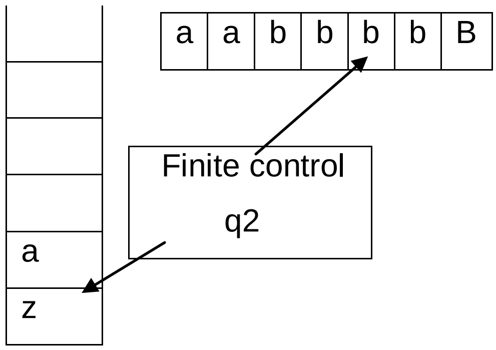
Step 2: a is pushed



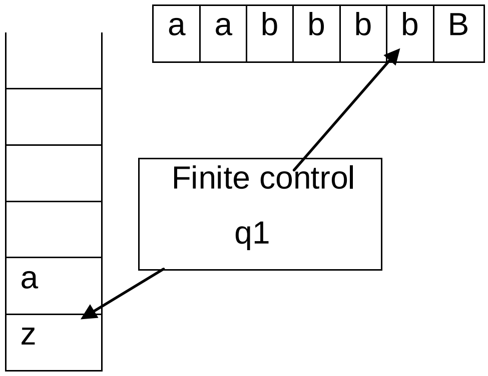
Step 3: a is pushed



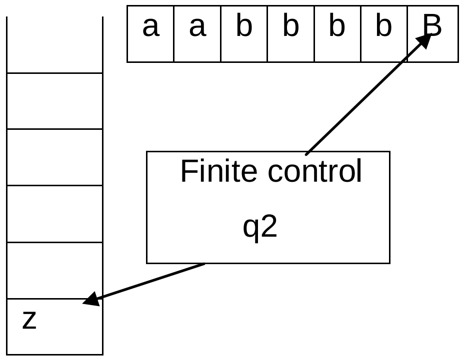
Step 4: read b no pop operation, state changes to q1



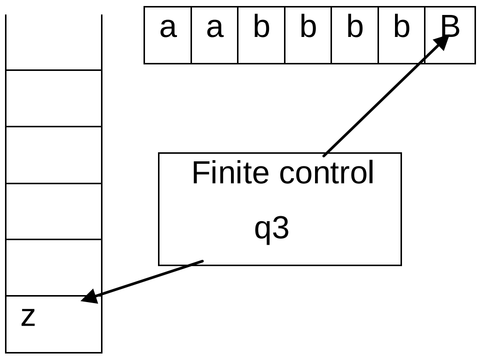
Step 5: read b, pop a, state changes to q2



Step 6: read b no pop operation, state Changes to q1



Step 7: read b, pop a, state changes to q2



Step 8: reaches the final state and halts

The PDA moves are as follows:-

$\delta(q0,a,z) = (q0,az)$

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

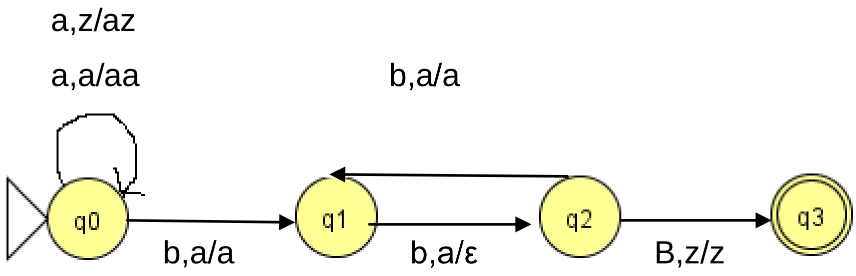
$\delta(q0,b,a) = (q1,aa)$

$\delta(q1,b,a) = (q2,\varepsilon)$

$\delta(q2,b,a) = (q1,a)$

$\delta(q1,b,a) = (q2,\varepsilon)$

$\delta(q2,B,z) = (q3,z)$



We will stimulate this for the following string aabbbb

δ (q0,aabbbb,z) $\vdash$ (q0,abbbb,az)

$\vdash$ (q0,bbbb,aaz)

$\vdash$ (q1,bbb,baaz)

$\vdash$ (q2,bb,az)

$\vdash$ (q1,b,baz)

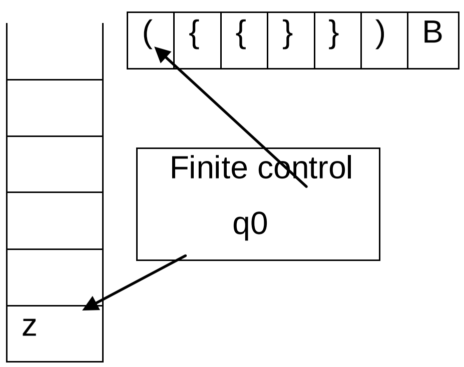
$\vdash$ (q2,ε,z)

$\vdash$ (q3,B)

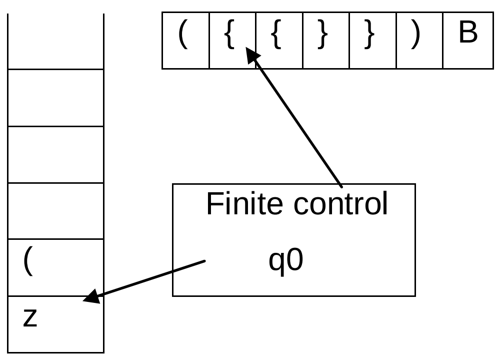
Example 4: Design a PDA which accepts the set of balanced parenthesis ({{()}})

Solution:

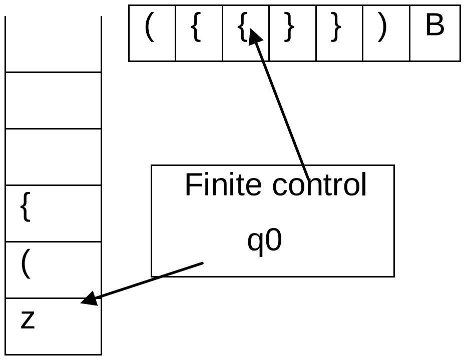
Read all open braces/parenthesis on to the stack, then whenever a closed parenthesis is seen, match it by popping off.



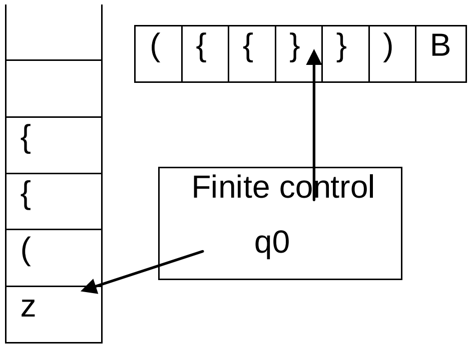
Step 1: Initial configuration



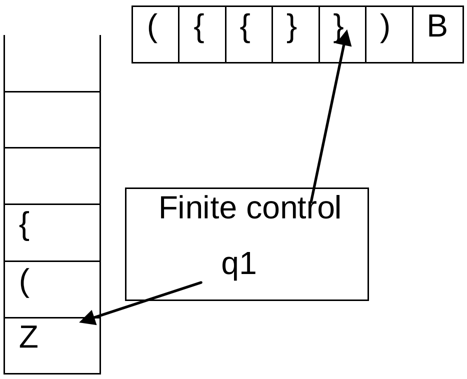
Step 2: push (



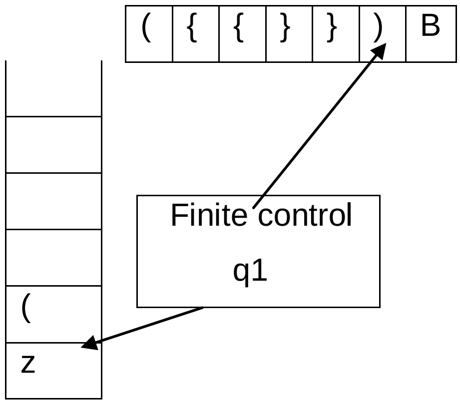
Step 3: push {



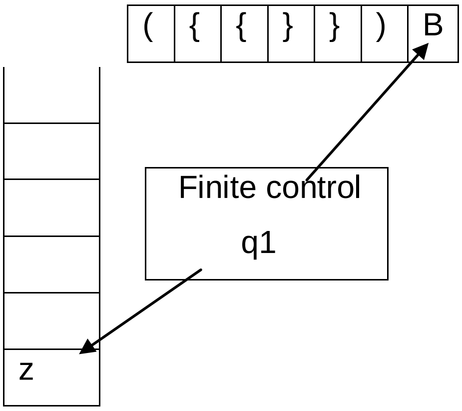
Step 4: push {



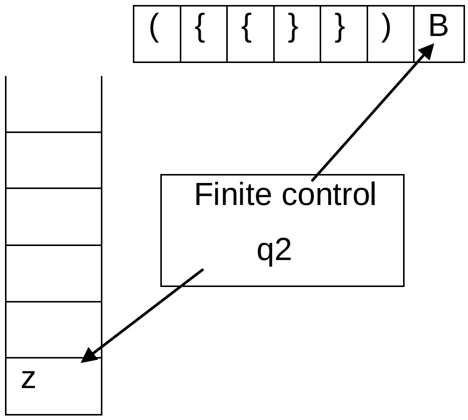
Step 5: pop { and change to state q1



Step 6: pop {



Step 7: pop (



Step 8: Reaches the final state q2

PDA moves:-

δ(q0,(,z) = (q0,(,z)

δ(q0,{,() = (q0,{()

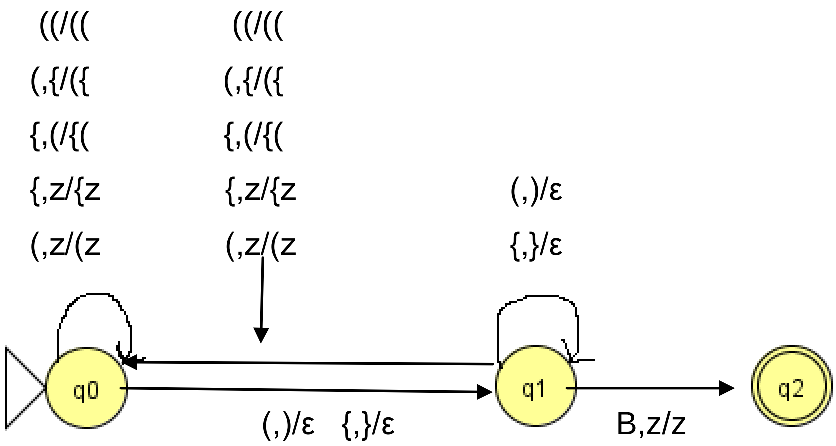
δ(q0,{,{) = (q0,{{)

δ(q0,},{) = (q1,ε)

δ(q1,},{) = (q1,ε)

δ(q1,),() = (q1,ε)

δ(q1,B) = (q2,z)



Example 5: Identify the language for a given PDA

M = ({q0,q1,q2}, {a, b, c}, {a, b, z}, δ, q0, z, {q2}). The mapping function is given as

δ(q0,a,z) = (q0,az)

δ(q0,b,z) = (q0,bz)

δ(q0,a,a) = (q0,aa)

δ(q0,b,a) = (q0,ba)

δ(q0,a,b) = (q0,ab)

δ(q0,b,b) = (q0,bb)

δ(q0,c,z) = (q1,z)

δ(q0,c,a) = (q1,a)

δ(q0,c,b) = (q1,b)

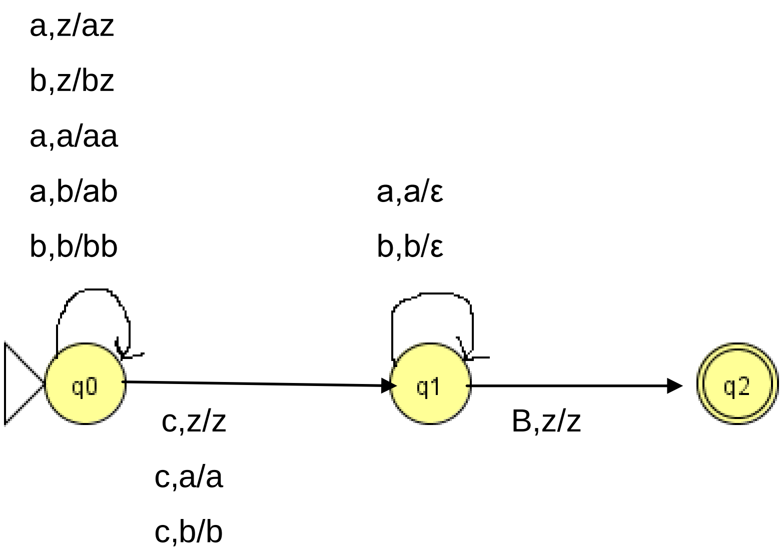
δ(q1,a,a) = (q1,ε)

δ(q1,b,b) = (q1,ε)

δ(q1,B,z) = (q2,z)

Simulate for the input bbacabb

Solution:



(q0,bbacabb,z) ├ (q0,bacabb,bz)
├ (q0,acabb,bbz)
├ (q0,cabb,abbz)
├ (q1,abb,abbz)
├ (q1,bb,bbz)
├ (q1,b,bz)
├ (q1,B,z)
├ (q2,z)

There is a language L(M) = $\{wcw^R / w \in (a+b)^*\}$ where w^R is reverse of w.

4.4. Equivalence of PDA and CFG

For a given PDA M = (Q, Γ, Σ, δ, q0, z, F). we will construct a grammar G such that L(G) = L(M)

- **Rule 1**

The production for the start symbol S are given by

S → [q0,z,q] for each state q in Q

- **Rule 2**

  For each move that pops a symbol from the stack with transition as

  δ(q,a,z) = (q1,ε) indicates a production as

  [q,z,q1] → a for q1 in Q.

- **Rule 3**

  For each move that does not pop a symbol from the stack with transition as

  δ(q,a,z) = (q1,z1,z2,z3,z4,,,,,,) induces a production as

  $[q,z,q_m]$ → a[q1,z1q2] [q2,z2q3] [q3,z3q4]

  [q4,z4q5],,,,,$[q_{m-1},z_m q_m]$ for each q_m in Q.

After defining all the rules, apply simplification of grammar to get the reduced grammar.

Example 1: Convert the given PDA to CFG.

P = ({q}, {i,e}, {x,z},δ,q,z,F) where δ is given by

δ(q,i,z) = (q,xz)

δ(q,e,x) = (q,ε)

δ(q,ε,z) = (q,ε)

Solution:

The non-nonterminals are {s,[qxq],[qzq]}

Equivalent productions are

S productions are given by rule 1:

S→[qzq]

The CFG for δ(q,i,z) = {(q,xz)} is obtained by rule 3

[q,z,q]→i[qxq][qzq]

The CFG for δ(q,e,x) = (q,ε) is obtained by rule 2

[q,x,q]→e

The CFG for δ(q,ε,z) = (q,ε) is obtained by rule 2

[qzq]→ε

If [qzq] is renamed as A and [qxq] is renamed to B. then

G = ({S,A,B}, {1,e}, P, S), where P is the production rule

S→A
A→iBA|ε
B→e

Convert the given PDA to CFG.

P = ({p,q}, {0,1}, {x,z},δ,q,z,F) where δ is given by

δ(q,1,z) = (q,xz)

δ(q,1,x) = (q,xx)

δ(q,ε,x) = (q,ε)

δ(q,0,x) = (p,x)

δ(p,1,x) = (p,ε)

δ(p,0,z) = (q,z)

Solution:

The non-nonterminals are

{s,[pxp],[pxq],[qxp],[qxq],[pzp],[pzq],[qzp],[qzq]}

S productions are given by rule 1:

S→[qzq] | [qzp]

For δ(q,1,z) = (q,xz) is obtained by rule 3

[qzq]→1[qxq][qzq]

[qzq]→1[qxp][pzq]

[qzp]→1[qxq][qzp]

[qzp]→1[qxp][pzp]

For δ(q,1,x) = (q,xx) is obtained by rule 3

[qxq]→1[qxq][qxq]

[qxq]→1[qxp][pxq]

[qxp]→1[qxq][qxp]

[qxp]→1[qxp][pxp]

For δ(q,ε,x) = (q,ε)

[qxq]→ε

For δ(q,0,x) = (p,x) is obtained by rule 3

[qxp]→0[pxp]

[qxp]→0[pxq]

For δ(p,1,x) = (p,ε) is obtained by rule 2

[pxp]→1

For δ(p,0,z) = (q,z) is obtained by rule 3

[pzq]→0[qzq]

[pzp]→0[qzp]

Renaming the variables we get

[qzq] to A, [qzp] to B, [pzq] to C, [pzp] to D, [qxq] to E, [qxp] to F, [pxp] to G and [pxq] to H.

G = ({S, A, B, C, D, E, F, G, H}, {0,1}, P, S)

S→A|B

A→1EA|1FC

B→!EB|1FD

C→0A

D→0B

E→1EE|1FH| ε|0H

F→1EF|1FG|0G

G→1

On simplifying the above grammar, we get

S→B
B→1EB|1FD
D→0
E→1EE| ε
F→1EF|1FG|0G
G→1

Convert the given CFG to PDA

I→a|b, S→aA, A→aABC|bB|a, B→b, C→c

Solution:

PDA will be M = (Q, Γ, Σ, δ, q0, z, F)

M = { Q = {q}, Σ = {a,b,c}, Γ = {a,b,c,S,A,B,C,I}, q0 = q, z = S, F}, where δ

is given by

δ(q, a, I) = (q,ε)

if there is only a single terminal on the right side place ε. i.e I→a

δ(q, b, I) = (q,ε)

if there is only a single terminal on the right side place ε. i.e I→b

$\delta(q, a, S) = (q,A)$

if there are nonterminals on the right hand side place that nonnonterminals.

i.e $S \rightarrow aA$

$\delta(q, a, A) = (q,ABC)$

if there are nonterminals on the right hand side place that nonnonterminals.

i.e $A \rightarrow aABC$

$\delta(q, b, A) = (q,B)$

if there are nonterminals on the right hand side place that nonnonterminals.

i.e $A \rightarrow bB$

$\delta(q, a, A) = (q,\varepsilon)$

if there is only a single terminal on the right side place ε. i.e $A \rightarrow a$

$\delta(q, b, B) = (q,\varepsilon)$

if there is only a single terminal on the right side place ε. i.e $B \rightarrow b$

$\delta(q, c, C) = (q,\varepsilon)$

if there is only a single terminal on the right side place ε. i.e $C \rightarrow c$

Example 3: Convert the given CFG to PDA

$S \rightarrow aBB$, $B \rightarrow bS|c$

Solution:

PDA will be M = $(Q, \Gamma, \Sigma, \delta, q0, z, F)$

M = { Q = {q}, $\Sigma = \{a,b,c\}$, Γ

given by

$\delta(q, a, S) = (q,BB)$

if there are nonterminals on the right hand side place that nonnonterminals.

i.e $S \rightarrow aBB$

$\delta(q, b, B) = (q,S)$

if there are nonterminals on the right hand side place that nonnonterminals.

i.e $B \rightarrow bS$

δ(q, c, B) = (q,ε)

if there is only a single terminal on the right side place ε. i.e B→c

Example 4: Convert the given CFG to PDA

S→aAA, A→aS|bS|a

Solution:

PDA will be M = (Q, Γ, Σ, δ, q0, z, F)

M = { Q = {q}, Σ ={a,b,c}, Γ ={a

given by

δ(q, a, S) = (q,AA)

if there are nonterminals on the right hand side place that nonnonterminals.

i.e S→aAA

δ(q, a, A) = (q,ε)

if there is only a single terminal on the right side place ε. i.e A→a

δ(q, a, A) = (q,S)

if there are nonterminals on the right hand side place that nonnonterminals.

i.e A→aS

δ(q, b, A) = (q,S)

if there are nonterminals on the right hand side place that nonnonterminals.

i.e A→bS

4.3. Types of PDA

Deterministic PDA (DPDA)

The PDA that has at most one choice of move in any state is called deterministic PDA. NPDA provides nondeterminism to a PDA. Deterministic PDA's (DPDA) are very useful in programming languages. For example parsers used in YACC are DPDA's.

A PDA P = (Q, Γ, Σ, δ, q0, z, F) is deterministic if and only if,

- ✓ δ(q,a,x) has at most one member for q ЄQ, aЄ
- ✓ δ(q,a,x) is not empty for some a Є

DPDA is less powerful than NPDA. The context free languages could be recognized by NPDA. **Therefore by default PDA is nondeterminism.**

- **Deterministic Context-Free language (DCFL)**
  - ✓ The language accepted by DPDA lies between regular language and context free language. This language is called deterministic context free language.
  - ✓ It is a subset of the language accepted by NPDA.
  - ✓ There are some limitations in DPDA, for example the languages for constructing a palindrome can be accepted by only NPDA.
  - ✓ The syntax of the programming languages is described by DCFL.
  - ✓ Deterministic Context-Free language is accepted by the grammar called as Context free grammar. In the compiler point of view it is represented as restricted grammar.
  - ✓ LR grammars are restricted grammars that generate DCFL.
  - ✓ LR stands for Left-Right scanning, rightmost derivation.

Closure Properties of DCFL

DCFL are closed under

- ✓ Union
- ✓ Concatenation
- ✓ Kleen's closure
- ✓ Complementation

=>**Note** CFL are not closed under complement but DCFL's are closed. Hence if the CFL is found to be closed under complement than it is a DCFL.

NORMAL FORMS FOR DPDA's

An y PDA is said to be deterministic, if it satisfies the following properties.

1. Whenever δ(q,a,X) is non-empty for some a in ∑, then δ(q,ε,X) is empty, and

2. For each q in Q, a in ∑ U {ε} and X in Γ, δ(q,a,X) contains at most one element.

The first rule prevent from considering the next input or ignoring it. Second rule prevents moving to two different ID's for same input symbol.

The DPDA can be represented in the normal form by imposing the following two lemmas on transition to restrict the operation.

- ✓ LEMMA 1:

  Every DCFFL is L(M) for a DPDA M=(Q,∑,τ)