PUSH DOWN AUTOMATA

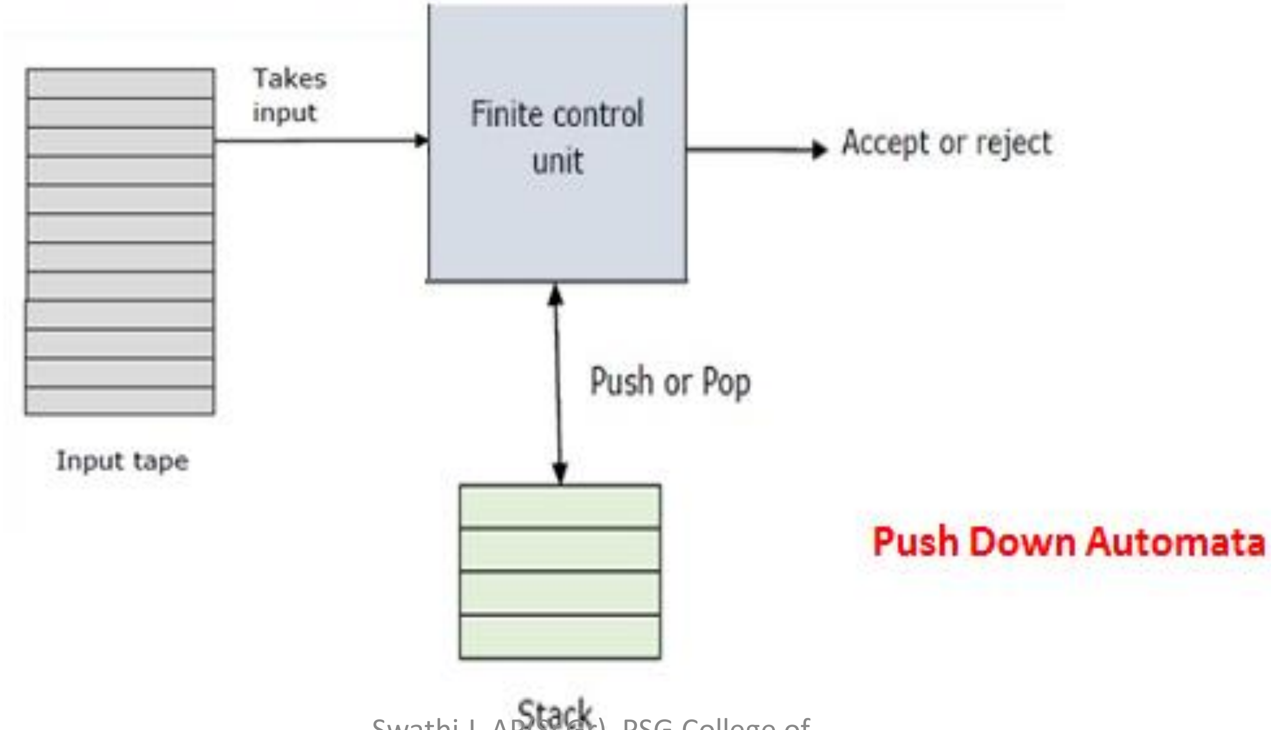
Swathi J

Assistant Professor(Sr.Gr.)

PSG College of Technology

Push Down Automata

- Memory in the form of finite stack (LIFO)
- Transition from one state to another is based on current state ,input and TOS only.



TRANSITIONS IN PDA

$(Q \times \Sigma \times \Gamma) \rightarrow Q \times \Gamma^*$

(q0 , a , z0) → (q1,a z0)

(q0 , a , z0) → (q1,z0)

(q0 , a , a) → (q1, ^)

ACTIONS PERFORMED ON STACK

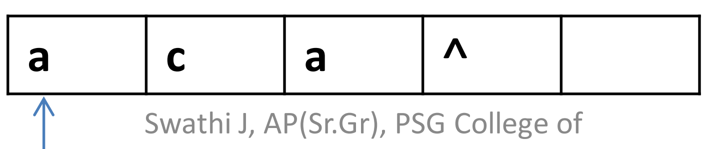
1.Push

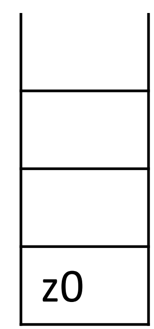
2. No Change

3. Pop

NOTE:

1. Empty Stack - z0
2. End of input symbol - ^



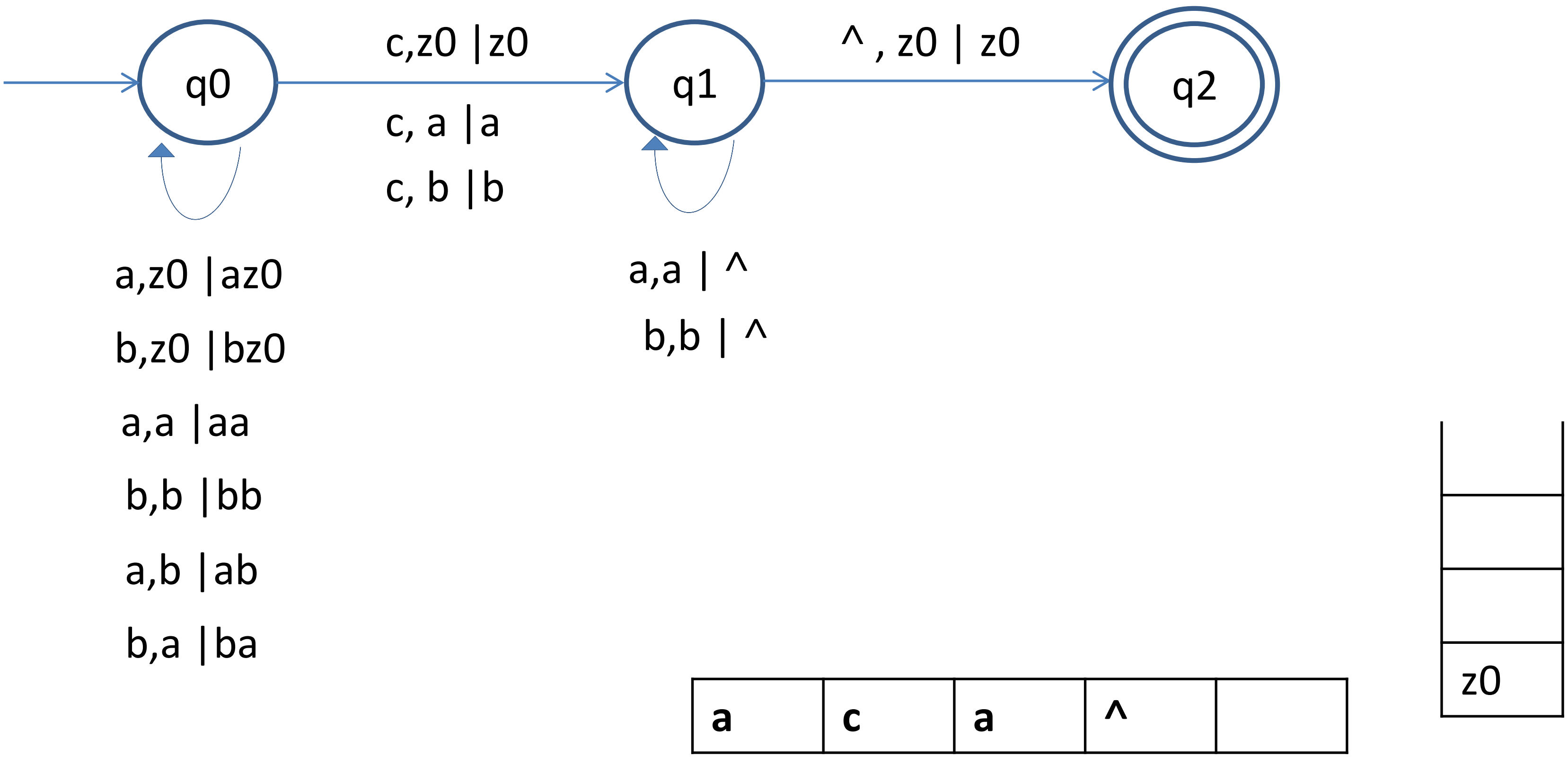


Let G be a CFG having the production

$S \rightarrow a\,S\,a \mid bSb \mid c$

L ={c, aca,bcb,aacaa, bbcbb,abcba,bacab……..}

$L = \{x\,c\,x^r \mid x \in \{a,b\}^*\}$



L ={x c x^r | x ϵ $\{a, b\} *$ }

STATE	INPUT	STACK SYMBOL	MOVES
q0	a	z0	(q0,a z0)
q0	b	z0	(q0,b z0)
q0	a	a	(q0,aa)
q0	b	b	(q0, bb)
q0	a	b	(q0,a b)
q0	b	a	(q0,ba)
q0	c	z0	(q1, z0)
q0	c	a	(q1,a)
q0	c	b	(q1, b)
q1	a	a	(q1, ^)
q1	b	b	(q1, ^)
q1	^	z0	(q1, z0)

Trace string : abcba

RESULTING STATE	UNREAD INPUT	STACK
q0	**a**bcba	z0
q0	**b**cba	a z0
q0	**c**ba	b a z0
q1	**b**a	b a z0
q1	**a** ^	a z0
q1	^	z0
q2	-	z0

Trace string : bcba

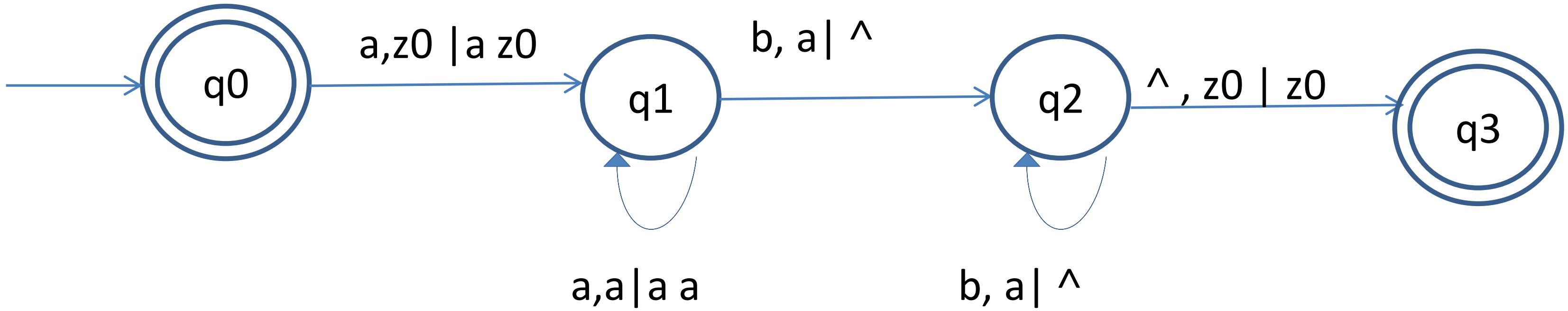
RESULTING STATE	UNREAD INPUT	STACK
q0	**b**cba	z0
q0	**c**ba	b z0
q1	**b**a	b z0
q1	**a**	z0

FORMAL DEFINITION OF PUSH DOWN AUTOMATA

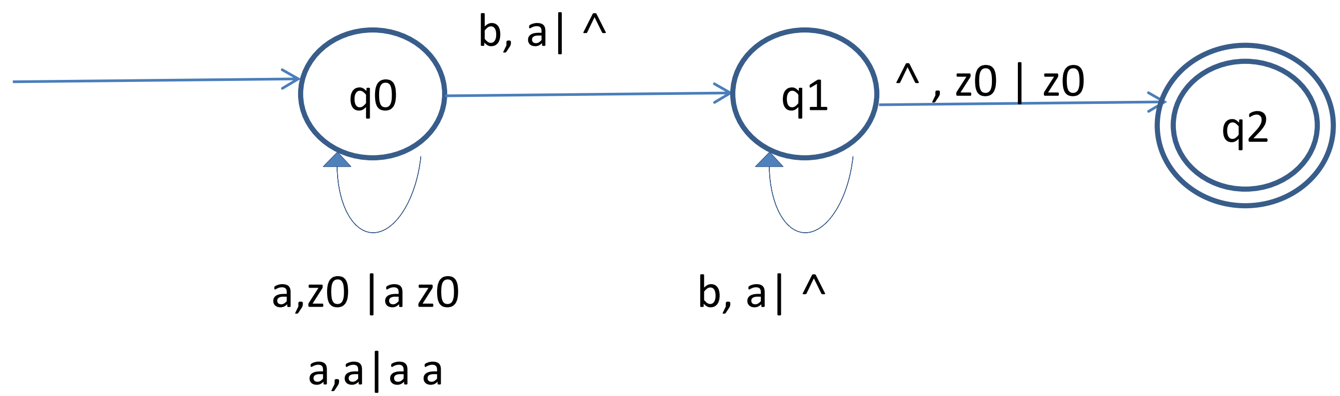
- A PDA is a **7 tuple (Q, Σ, Γ , q0, z0, F, δ)** where
  - Q – finite set of states
  - Σ – finite set of input symbols
  - Γ – finite set of stack symbols
  - q0- start state   q0 ϵ Q
  - Z0- initial stack symbol, q0 ϵ Γ
  - F - set of final states F ⊆ Q

-- δ – transition function **(Q x Σ x Γ) → Q x Γ***

Design a PDA to accept the language L(G) = { $a^n b^n$ | n ≥ 0 }



Design a PDA to accept the language L(G) = { $a^n b^n$ | n > 0 }



TYPES

1. DPDA- Deterministic PDA
2. NPDA- Non Deterministic PDA

DPDA- Deterministic PDA

Let M= (Q, Σ, Γ , q0, z0, F, δ) be a PDA. M is deterministic if it satisfies both the following conditions

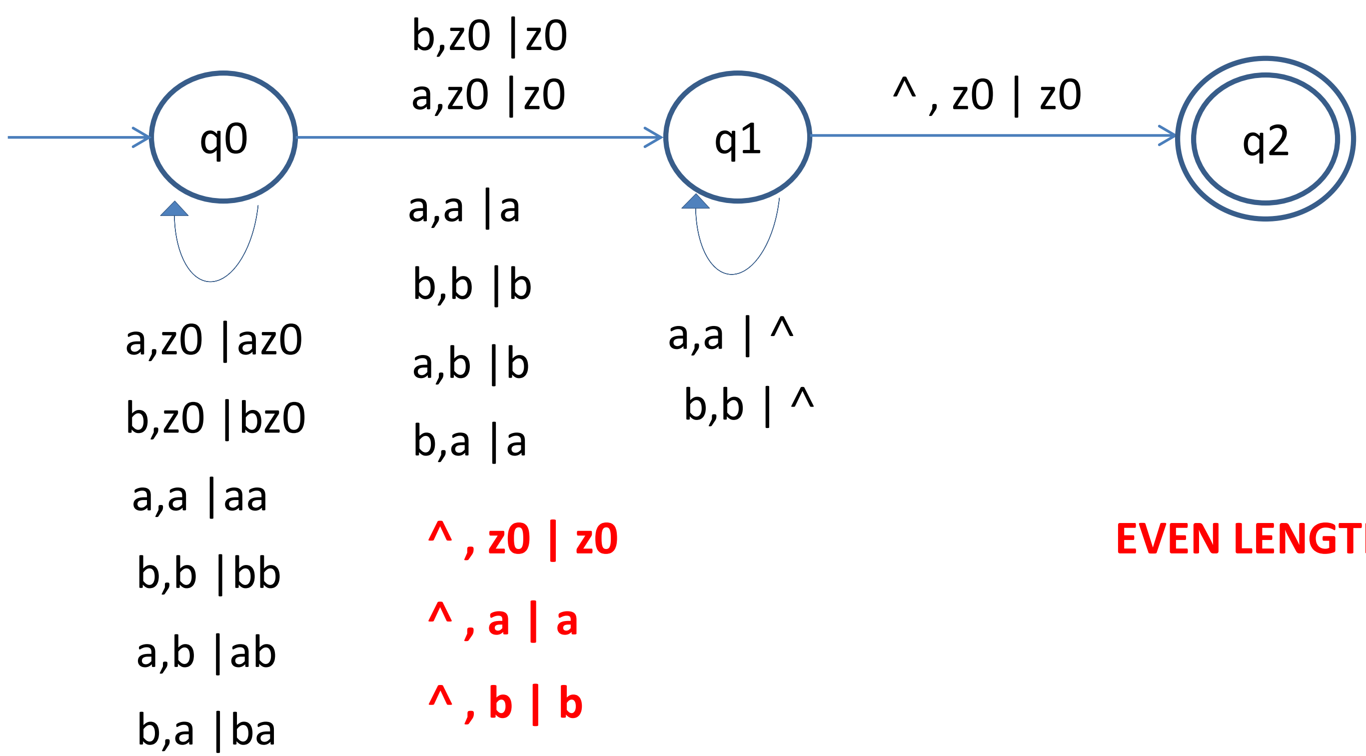
1. For any $q \in Q$, $a \in \Sigma \cup \{\wedge\}$ and $X \in \Gamma$, the set (q,a,X) has at most one element.

2. For any $q \in Q$ and $X \in \Gamma$, if (q, ^,X) ≠ NULL then (q,a,X) = NULL for every $a \in \Sigma$

POWER OF DPDA ≠ POWER OF NPDA

PDA to accept palindromes – odd length

Strings={a,b, aaa,bbb,aba,bab,aabaa,ababa…….}

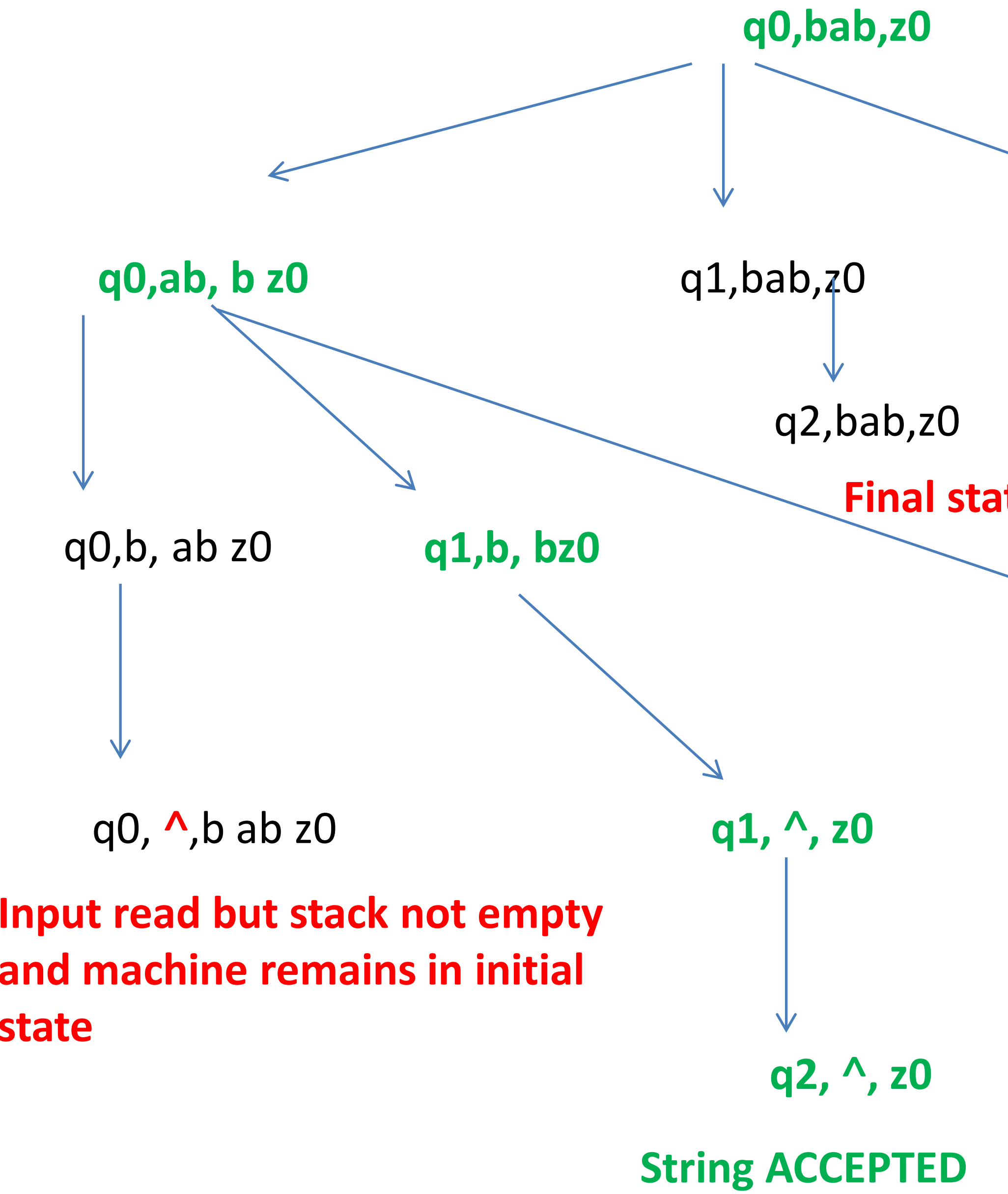


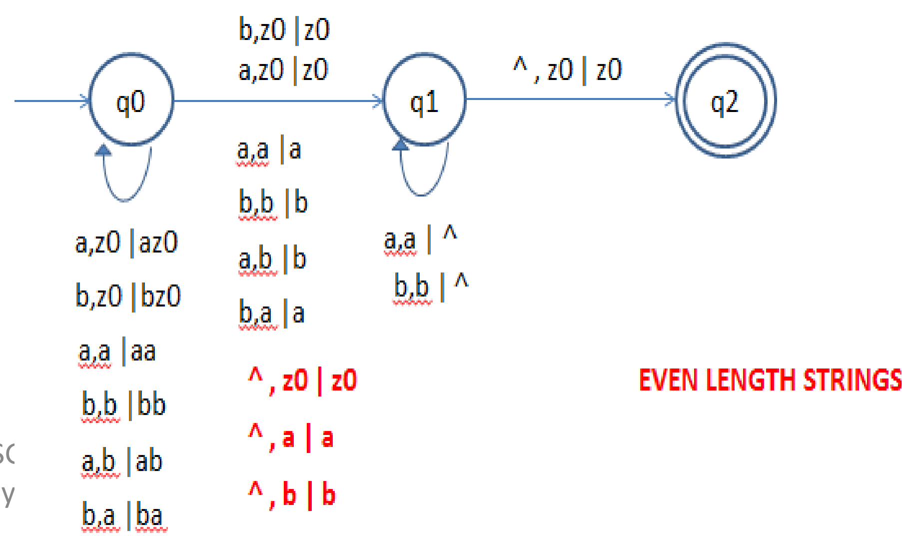
EVEN LENGTH STRINGS

PDA to accept palindromes – odd & even length

Strings={^ ,a,b, aa,bb, aaa,bbb,aba,bab, aaaa,abba, aabaa,ababa…….}

COMPUTATION TREE : bab





COMPUTATION TREE : aa

Draw PDA for the language L(G) = { $a^n b\, a^n$ | n ≥ 1 }

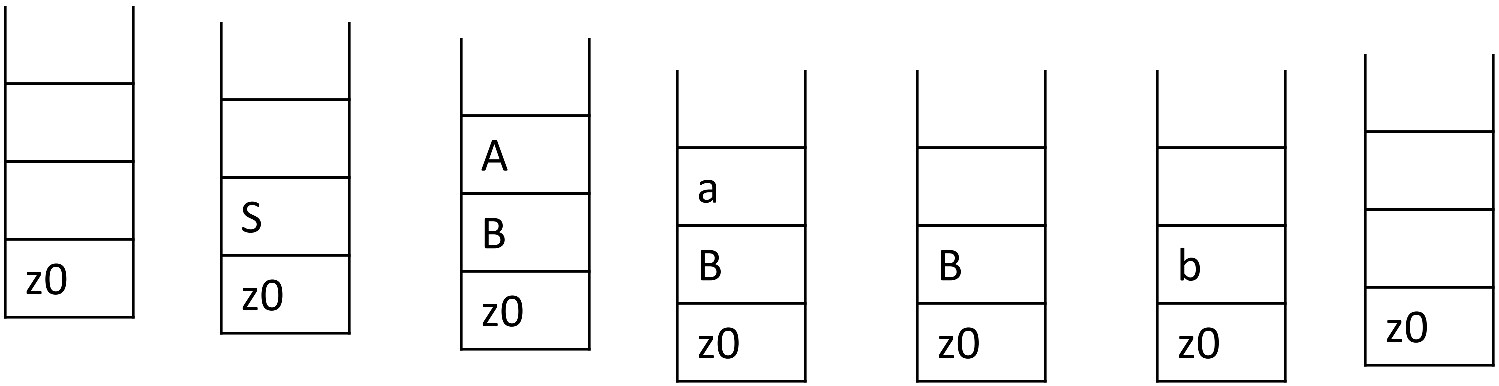
Draw a PDA for the language L(G) = { $a^n b^{n+1}$ | n ≥ 0 }

CFG TO PDA

1. Top down PDA
2. Bottom up PDA

TOP DOWN PDA CORRESPONDING TO CFG

S → AB
A → a
B → b



TO IMPLEMENT TOP DOWN APPROACH USING STACK

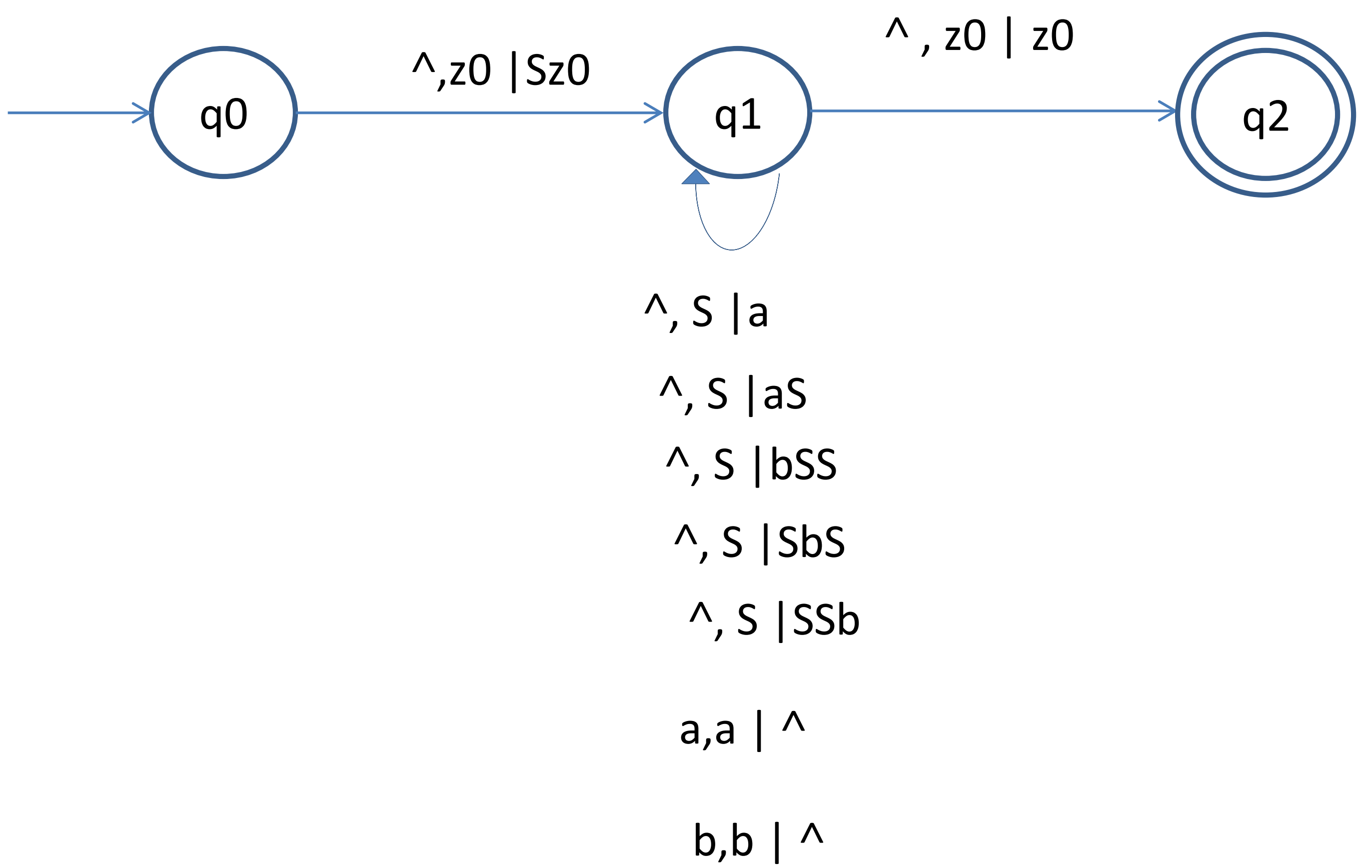
1. Push start symbol into stack

2. Replace variable on TOS with production on RHS (Push production in reverse order)

3. Pop symbols from stack if it matches with input symbol

Construct a TOP DOWN PDA to accept strings with more number of a’s than b’s

S → a |aS| bSS | SbS | SSb



S → a |aS| bSS | SbS | SSb

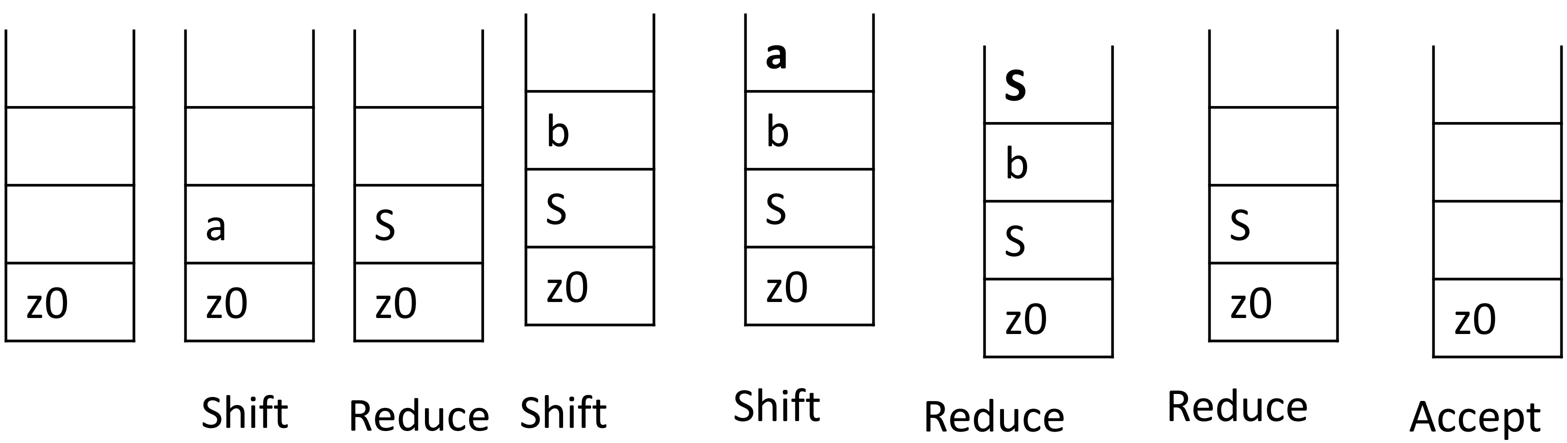
STATE	INPUT	STACK SYMBOL	MOVES
q0	^	z0	(q1, Szo)
q1	^	S	(q1, a) (q1,aS) (q1, bSS) (q1, SbS) (q1,SSb)
q1	a	a	(q1, ^)
q1	b	b	(q1, ^)
q1	^	z0	(q2, zo)

BOTTOM UP PDA CORRESPONDING TO CFG

S → a |aS| bSS | SbS | SSb

String: aba

aba => Sba => SbS => S



$S \rightarrow a \mid aS \mid bSS \mid SbS \mid SSb$

S → a |aS| bSS | SbS | SSb

	STATE	INPUT	STACK SYMBOL	MOVES
SHIFT	q	a	X	(q,aX)
S→ a	q	^	a	(q,S)
	q	^	S	(q1, ^)
S→ aS	q1	^	a	(q, S)
	q	^	S	(q2, ^)
S→ bSS	q2	^	S	(q3, ^)
	q3	^	b	(q, S)
	q	^	b	(q4, ^)
S→ SSb	q4	^	S	(q5, ^)
	q5	^	S	(q, S)
	q	^	S	(q6, ^)
S→ SSb	q6	^	b	(q7, ^)
	q7	^	S	(q, S)
	q	^	S	(qs, ^)
ACCEPT	qs	^	z0	(qf, z0)

PDA FOR THE LANGUAGE L={$0^N 1^N 0^N$ | N > 0}

PUMPING LEMMA FOR CONTEXT FREE LANGUAGE

- USED TO PROVE THAT A LANGUAGE IS **NOT** CONTEXT FREE.
- Let L be a CFL. Then there exist a constant "n" such that for every string u in L such that |u| ≥ n we can break u into 5 strings vwxyz such that
  - |wy| > **0**
  - |wxy| ≤ n
  - For all m ≥ 0, the string v w^m x y^m z is in language L

THE LANGUAGE L={$0^N 1^N 0^N$ | N > 0} is NOT regular

n=6 (pumping constant)

u= 000111000

v= 00 w=01 x= 11 y=00 z= 0

|wy| > 0
|wxy| ≤ n
For all m ≥ 0, the string v w^m x y^m z is in language

v w^0 x y^0 z = 00 11 0 → invalid string that does NOT belongs to language

L is not CFL.