PUSH DOWN AUTOMATA

ITEU133

AUTOMATA AND THEORY OF COMPUTATION

Greibach Normal Form

Greibach Normal Form (GNF), restriction are put on the positions in which terminals and variables can appear.

GNF is useful in simplifying some proofs and making constructions such as Push Down Automaton (PDA) accepting a CFG.

Greibach Normal Form

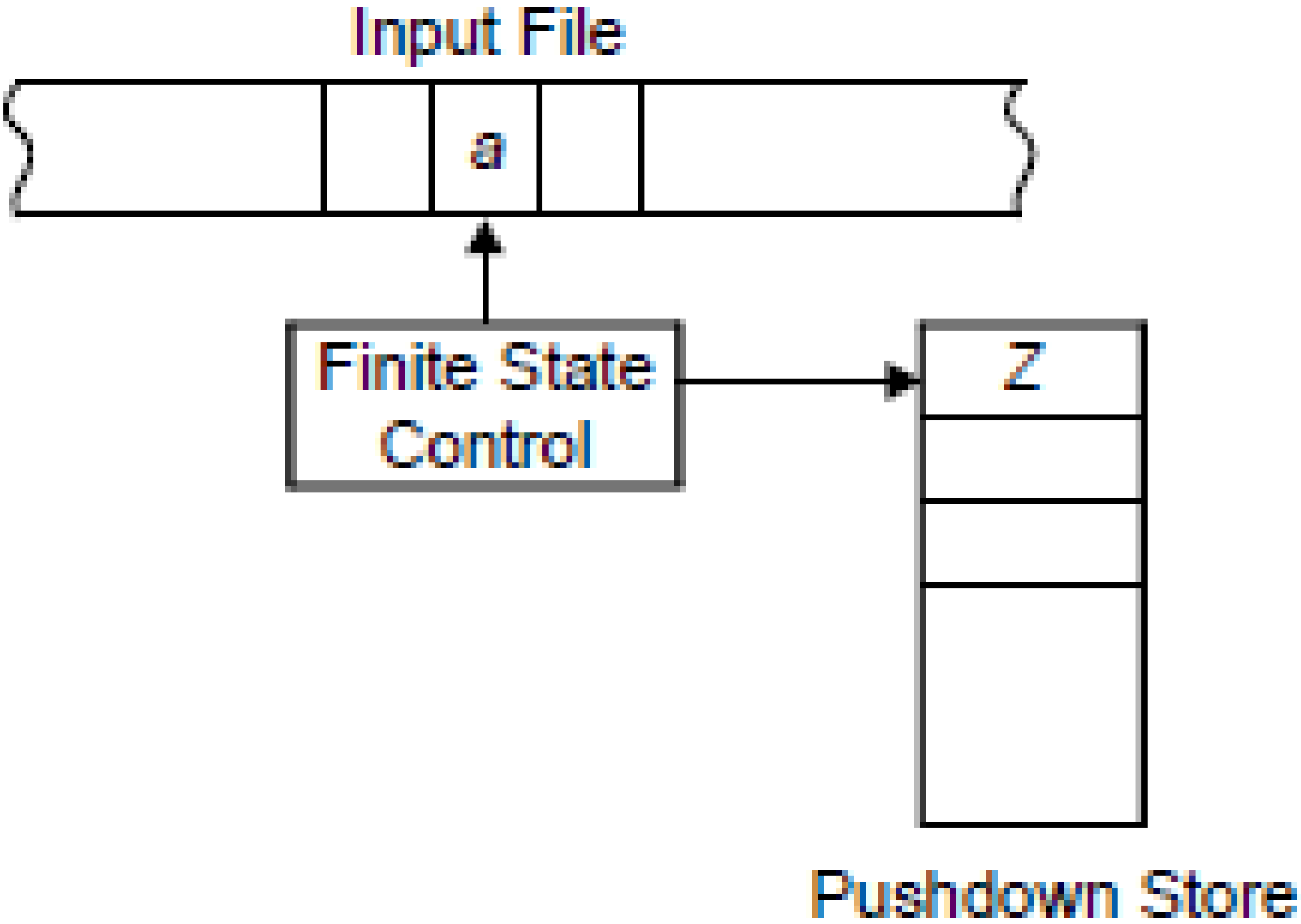
$S \rightarrow abSb \mid aa$

PUSHDOWN AUTOMATA (PDA)

Informally:

- A PDA is an NFA-ε with a stack.
- Transitions are modified to accommodate stack operations.

- **Questions:**
  - What is a stack?
  - How does a stack help?
- A DFA can "remember" only a finite amount of information, whereas a PDA can "remember" an infinite amount of (certain types of) information.


Input File
a
Finite State Control
Z
Pushdown Store

PUSHDOWN AUTOMATA (PDA)

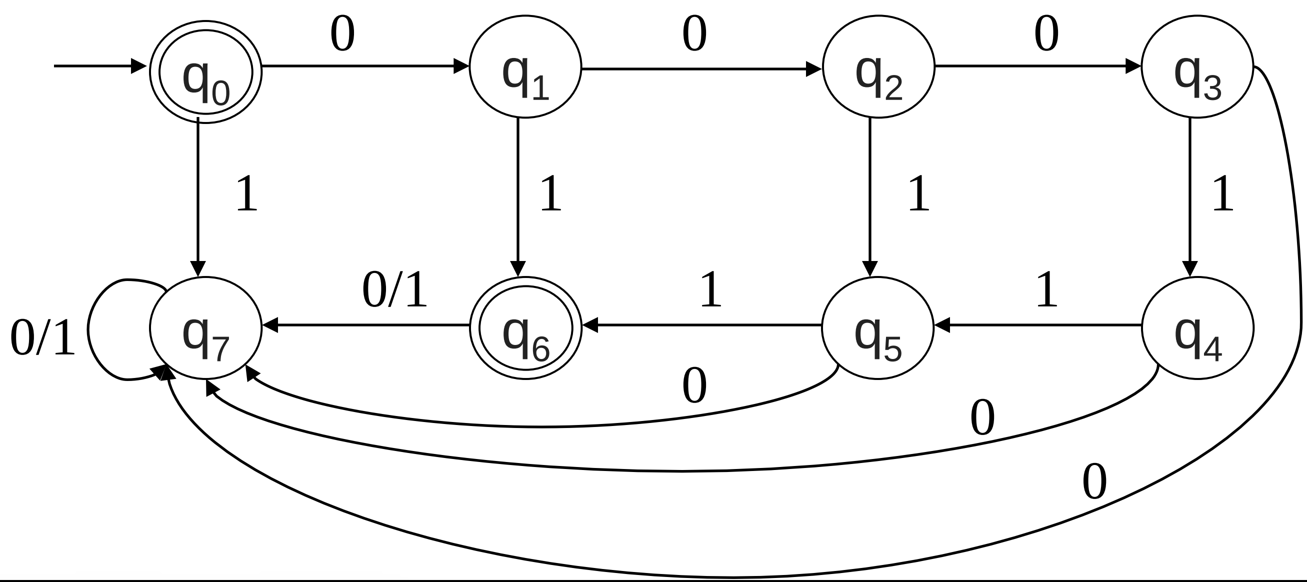
Example:

$\{0^n1^n \mid 0 =< n\}$ is *not* regular, but

$\{0^n1^n \mid 0 \leq n \leq k$, for some fixed $k\}$ is regular, for any fixed k.

For k=3:

$L = \{\varepsilon, 01, 0011, 000111\}$



PUSHDOWN AUTOMATA (PDA)

- In a DFA, each state remembers a finite amount of information.
- To get $\{0^n1^n \mid 0 \leq n\}$ with a DFA would require an infinite number of states using the preceding technique.
- An infinite stack solves the problem for $\{0^n1^n \mid 0 \leq n\}$ as follows:
  - Read all 0's and place them on a stack
  - Read all 1's and match with the corresponding 0's on the stack
- Only need two states to do this in a PDA
- Similarly for $\{0^n1^m0^{n+m} \mid n,m \geq 0\}$

FORMAL DEFINITION OF A PDA

- A pushdown automaton (PDA) is a seven-tuple:

  $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

  Q A finite set of states

  Σ A finite input alphabet

  Γ A finite stack alphabet

  q_0 The initial/starting state, q_0 is in Q

  z_0 A starting stack symbol, is in Γ

  F A set of final/accepting states, which is a subset of Q

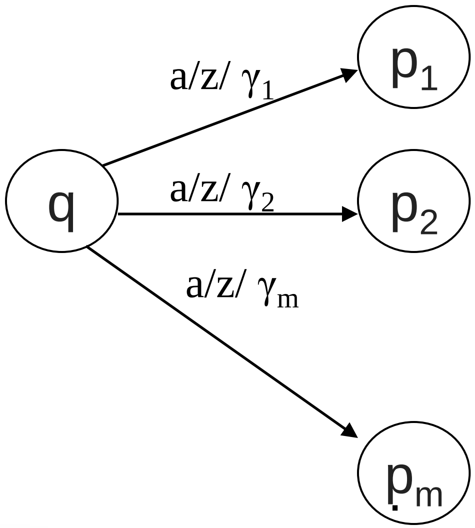
  δ A transition function, where

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

TWO TYPES OF PDA TRANSITIONS:

$\delta(q, a, z) = \{(p_1,\gamma_1), (p_2,\gamma_2),\ldots, (p_m,\gamma_m)\}$

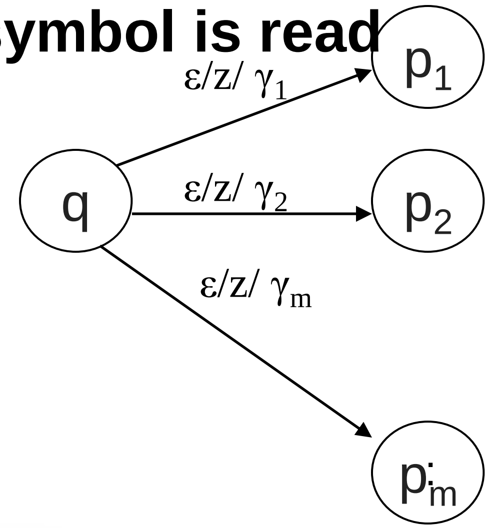
- Current state is q
- Current input symbol is a
- Symbol currently on top of the stack z
- Move to state p_i from q
- Replace z with γ_i on the stack (leftmost symbol on top)
- Move the input head to the next input symbol



TWO TYPES OF PDA TRANSITIONS:

$(q, \varepsilon, z) = \{(p_1,\gamma_1), (p_2,\gamma_2),\ldots, (p_m,\gamma_m)\}$

- Current state is q
- Current input symbol is not considered
- Symbol currently on top of the stack z
- Move to state p_i from q
- Replace z with γ_i on the stack (leftmost symbol on top)
- **No input symbol is read**



Example:

Q={s,q,f}   Σ = {a, b}   Γ = {a, b, C}   F={f}

δ

1	**((S,ε, ε), (q, C))**	**5**	**((q,b,c), (q, bC))**
2	((q, a, C), (q, aC))	6	((q,b,b),(q,bb))
3	((q,a,a),(q,aa))	7	((q,b,a), (q, ε))
4	((q,a,b),(q, ε))	8	((q, ε,C),(f, ε))

W= a b b b a a b a

Example #2

Q={s,q,f}   Σ = {a, b}   Γ = {a, b, C}   F={f}

δ

1	((S,a, ε), (q, C))	5	((q,a,b),(q, ε))
2	((S, b, ε), (q, b))	6	((q,b,a),(q, ε))
3	((q,a,a),(q,aa))	7	((q, ε, ε), (f, ε))
4	((q,b,b), (q,bb))		

W= a b b b a a b a

DRAW A TRANSITION DIAGRAM# 3

$$(Q = \{q_0, q_1, q_2, q_3\}, \Sigma = \{a, b\}, \Gamma = \{0, 1\}, \delta, q_0, Z = 0, F = \{q_3\})$$

where

$$\delta(q_0, a, 0) = \{(q_1, 10), (q_3, \lambda)\}$$
$$\delta(q_0, \lambda, 0) = \{(q_3, \lambda)\}$$
$$\delta(q_1, a, 1) = \{(q_1, 11)\}$$
$$\delta(q_1, b, 1) = \{(q_2, \lambda)\}$$
$$\delta(q_2, b, 1) = \{(q_2, \lambda)\}$$
$$\delta(q_2, \lambda, 0) = \{(q_3, \lambda)\}$$

An Example of NPDA Execution

Let us consider the NPDA given by

$$\delta(q_0, a, 0) = \{(q_1, 10), (q_3, \lambda)\}$$
$$\delta(q_0, \lambda, 0) = \{(q_3, \lambda)\}$$
$$\delta(q_1, a, 1) = \{(q_1, 11)\}$$
$$\delta(q_1, b, 1) = \{(q_2, \lambda)\}$$
$$\delta(q_2, b, 1) = \{(q_2, \lambda)\}$$
$$\delta(q_1, \lambda, 0) = \{(q_3, \lambda)\}$$

It is possible for us to recognize the string "***aaabbb***" using the following sequence of "Moves":

Example: (balanced parentheses)

$M = (\{q_1\}, \{“(“, “)”\}, \{L, \#\}, \delta, q_1, \#, \varnothing)$

δ:

(1) $\delta(q_1, (, \#) = \{(q_1, L\#)\}$

(2) $\delta(q_1,), \#) = \varnothing$ // illegal, string rejected

(3) $\delta(q_1, (, L) = \{(q_1, LL)\}$

(4) $\delta(q_1,), L) = \{(q_1, \varepsilon)\}$

(5) $\delta(q_1, \varepsilon, \#) = \{(q_1, \varepsilon)\}$ // if no character read, & stack hits bottom

(6) $\delta(q_1, \varepsilon, L) = \varnothing$ // illegal, string rejected

Goal: (acceptance)

Terminate in a non-null state

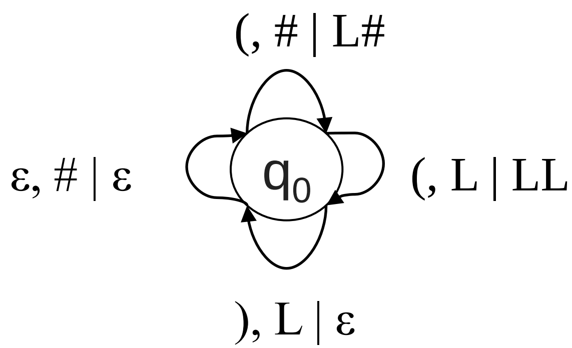
Read the entire input string

Terminate with an empty stack

Informally, a string is accepted if there exists a computation that uses up all the input and leaves the stack empty.

How many rules should be in delta?

Transition Diagram:



Example Computation:

Current Input	**Stack**	**Transition**
(())	#	
())	L#	(1)- Could have applied rule (5), but
))	LL#	(3) it would have done no good
)	L#	(4)
ε	#	(4)
ε	-	(5)

- **Example PDA #6:** For the language {x | x = wcw^r and w in {0,1}*, but sigma={0,1,c}} w=1c1

  $M = (\{q_1, q_2\}, \{0, 1, c\}, \{R, B, G\}, \delta, q_1, R, \emptyset)$

  δ:

  (1) $\delta(q_1, 0, R) = \{(q_1, BR)\}$

  (2) $\delta(q_1, 0, B) = \{(q_1, BB)\}$

  (3) $\delta(q_1, 0, G) = \{(q_1, BG)\}$

  (4) $\delta(q_1, c, R) = \{(q_2, R)\}$

  (5) $\delta(q_1, c, B) = \{(q_2, B)\}$

  (6) $\delta(q_1, c, G) = \{(q_2, G)\}$

  (7) $\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}$

  (8) $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$

  (9) $\delta(q_1, 1, R) = \{(q_1, GR)\}$

  (10) $\delta(q_1, 1, B) = \{(q_1, GB)\}$

  (11) $\delta(q_1, 1, G) = \{(q_1, GG)\}$

  (12) $\delta(q_2, 1, G) = \{(q_2, \varepsilon)\}$

- **Notes:**
  - Only rule #8 is non-deterministic.
  - Rule #8 is used to pop the final stack symbol off at the end of a computation.

- **Example PDA #7:** For the language $\{x \mid x = wcw^r \text{ and } w \text{ in } \{0,1\}^*$, but sigma=$\{0,1,c\}\}$

$$M = (\{q_1, q_2\}, \{0, 1, c\}, \{R, B, G\}, \delta, q_1, R, \emptyset)$$

δ:

(1) $\delta(q_1, 0, R) = \{(q_1, BR)\}$

(2) $\delta(q_1, 0, B) = \{(q_1, BB)\}$

(3) $\delta(q_1, 0, G) = \{(q_1, BG)\}$

(4) $\delta(q_1, c, R) = \{(q_2, R)\}$

(5) $\delta(q_1, c, B) = \{(q_2, B)\}$

(6) $\delta(q_1, c, G) = \{(q_2, G)\}$

(7) $\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}$

(8) $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$

(9) $\delta(q_1, 1, R) = \{(q_1, GR)\}$

(10) $\delta(q_1, 1, B) = \{(q_1, GB)\}$

(11) $\delta(q_1, 1, G) = \{(q_1, GG)\}$

(12) $\delta(q_2, 1, G) = \{(q_2, \varepsilon)\}$

w=**0**1c10

Example PDA #2: For the language $\{x \mid x = ww^r \text{ and } w \text{ in } \{0,1\}^*\}$

$M = (\{q_1, q_2\}, \{0, 1\}, \{R, B, G\}, \delta, q_1, R, \varnothing)$

00 00 00

δ:

(1) $\delta(q_1, 0, R) = \{(q_1, BR)\}$

(2) $\delta(q_1, 1, R) = \{(q_1, GR)\}$

(3) $\delta(q_1, 0, B) = \{(q_1, BB), (q_2, \varepsilon)\}$

(4) $\delta(q_1, 0, G) = \{(q_1, BG)\}$

(5) $\delta(q_1, 1, B) = \{(q_1, GB)\}$

(6) $\delta(q_1, 1, G) = \{(q_1, GG), (q_2, \varepsilon)\}$

(7) $\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}$

(8) $\delta(q_2, 1, G) = \{(q_2, \varepsilon)\}$

(9) $\delta(q_1, \varepsilon, R) = \{(q_2, \varepsilon)\}$

(10) $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$

Notes:

– Rules #3 and #6 are non-deterministic.

– Rules #9 and #10 are used to pop the final stack symbol off at the end of a computation.

EXAMPLE COMPUTATION:

(1) $\delta(q_1, 0, R) = \{(q_1, BR)\}$

(2) $\delta(q_1, 1, R) = \{(q_1, GR)\}$

(3) $\delta(q_1, 0, B) = \{(q_1, BB), (q_2, \varepsilon)\}$

(4) $\delta(q_1, 0, G) = \{(q_1, BG)\}$

(5) $\delta(q_1, 1, B) = \{(q_1, GB)\}$

(6) $\delta(q_1, 1, G) = \{(q_1, GG), (q_2, \varepsilon)\}$

(7) $\delta(q_2, 0, B) = \{(q_2, \varepsilon)\}$

(8) $\delta(q_2, 1, G) = \{(q_2, \varepsilon)\}$

(9) $\delta(q_1, \varepsilon, R) = \{(q_2, \varepsilon)\}$

(10) $\delta(q_2, \varepsilon, R) = \{(q_2, \varepsilon)\}$

State	Input	Stack	Rule Applied	
q_1	010010	R		
q_1	10010	BR	(1)	From (1) and (9)
q_1	0010	GBR	(5)	
q_1	010	BGBR	(4)	
q_2	10	GBR	(3) option #2	
q_2	0	BR	(8)	
q_2	ε	R	(7)	
q_2	ε	ε	(10)	

EXERCISES:

– 0011001100
– 011110
– 0111

FORMAL DEFINITIONS FOR PDAS

- Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ be a PDA.
- **Definition:** An *instantaneous description* (ID) is a triple (q, w, γ), where q is in Q, w is in Σ^* and γ is in Γ^*.
  - q is the current state
  - w is the unused input
  - γ is the current stack contents
- **Example:** (for PDA #2)

$(q_1, 111, GBR)$	$(q_1, 11, GGBR)$
$(q_1, 111, GBR)$	$(q_2, 11, BR)$
$(q_1, 000, GR)$	$(q_2, 00, R)$

- Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ be a PDA.
- **Definition:** Let a be in $\Sigma \cup \{\varepsilon\}$, w be in Σ^*, z be in Γ, and α and β both be in Γ^*. Then:

$$(q, aw, z\alpha) \vdash_M (p, w, \beta\alpha)$$

  if $\delta(q, a, z)$ contains (p, β).
- Intuitively, if I and J are instantaneous descriptions, then I $\vdash$ J means that J follows from I by one transition.

- **Examples:** (PDA #2)

$(q_1, 111, GBR) \vdash (q_1, 11, GGBR)$ (6) option #1, with a=1, z=G, β=GG, w=11, and α= BR

$(q_1, 111, GBR) \vdash (q_2, 11, BR)$ (6) option #2, with a=1, z=G, β= ε, w=11, and α= BR

$(q_1, 000, GR) \vdash (q_2, 00, R)$ Is *not* true, For any a, z, β, w and α

- **Examples:** (PDA #1)

$(q_1, (())), L\#) \vdash (q_1, ())),LL\#)$ (3)

- **Example #2:** Consider the following CFG in GNF.

  (1) S –> aA

  (2) S –> aB

  (3) A –> aA

  (4) A –> aB

  (5) B –> bB

  (6) B –> b

  G is in GNF

  L(G) = a+b+

  Construct M as:

  Q = {q}

  Σ = T = {a, b}

  Γ = V = {S, A, B}

  z = S

  (1) δ(q, a, S) = {(q, A), (q, B)} From productions #1 and 2, S->aA, S->aB

  (2) δ(q, a, A) = {(q, A), (q, B)} From productions #3 and 4, A->aA, A->aB

  (3) δ(q, a, B) = Ø

  (4) δ(q, b, S) = Ø

  (5) δ(q, b, A) = Ø

  (6) δ(q, b, B) = {(q, B), (q, ε)} From productions #5 and 6, B->bB, B->b

  (7) δ(q, ε, S) = Ø

  (8) δ(q, ε, A) = Ø

  (9) δ(q, ε, B) = Ø

  Recall δ: Q x (Σ U {ε}) x Γ –> finite subsets of Q x Γ