Module - 3

Context - Free Grammars.

A grammar $G = (V, T, P, S)$ is said to be type 2 grammar or context-free grammar if all the productions are of the form $A \rightarrow \alpha$ where $\alpha \in (V \cup T)^*$ and A is non-terminal.

The language generated from this grammar is called context free language. Pushdown automata (PDA) can be constructed to recognize the language generated from this grammar.

Designing Context - Free Grammars.

Rules.

- If L has the property that every string in it has two regions and those regions must bear some relationship to each other (such as being of the same length) then the two regions must be generated in tandem. Otherwise, there is no way to enforce the necessary constraint.
- To generate a string with multiple regions that must occur in some fixed

order but do not have to correspond to each other, use a rule of the form

$A \rightarrow BC \ldots$

- To generate a string with two regions that must occur in some fixed order and that must correspond to each other, start at the outside edges of the string and generate towards the middle.

Q) Design a context-free grammar for $L = \{a^n b^n c^m : n, m \geq 0\}$. Let $G = (\{S, N, C, a, b, c\}, \{a, b, c\}, R, S\}$

$$R = \{ S \rightarrow NC,\ N \rightarrow aNb,\ N \rightarrow \varepsilon,\ C \rightarrow cC,\ C \rightarrow \varepsilon \}$$

Q) Design a context-free grammar for $L = \{a^n b^n : n \geq 0\}$. Let $G = (\{S, M, a, b\}, \{a, b\}, R, S\}$

$$R = \{ S \rightarrow MS,\ S \rightarrow \varepsilon,\ M \rightarrow aMb,\ M \rightarrow \varepsilon \}$$

Simplifying Context-free Grammars

Simplifying CFG or eliminating useless symbols and productions.

Q1) Eliminate useless symbols in the grammar

$S \rightarrow aA \mid bB$

$A \rightarrow aA \mid a$

$B \rightarrow bB$

$D \rightarrow ab \mid Ea$

$E \rightarrow aC \mid d.$

Step 1 :

ov	nv	productions.
ϕ	A, D, E	$A \rightarrow a$ $D \rightarrow ab$ $E \rightarrow d$ /
A, D, E	A, D, E, S	$S \rightarrow aA$ $A \rightarrow aA$ $D \rightarrow Ea$
AD, E, S	AD, E, S	-

Step 2 :

P'	T'	V'
-	-	S
$S \rightarrow aA$	a	S, A
$A \rightarrow a \mid aA$	a	S, A

$G' = V', T', P', S'$

$V' = \{S, A\}$

$T' = \{a\}$

$P' = \{ S \to aA$

$A \to a \mid aA$

$\}$

S is start symbol.

Q2) Simplify the following CFG

$S \to aA \mid a \mid Bb \mid \epsilon C$

$A \to aB$

$B \to a \mid Aa$

$C \to \epsilon CD$

$D \to ddd$

Step 1:

OV	NV	productions
ϕ	S, B, D	$S \to a$ $B \to a$ $D \to ddd$
S, B, D	S, B, D, A	$S \to Bb$ $A \to aB$
S, B, D, A	S, B, D, A	$S \to aA$ $B \to Aa$

Step2:

P'	T'	V'
- -	-	S
$S \to a/Bb/Aa$	a, b	S, A, B
$A \to aB$	a, b	S, A, B
$B \to a/Aa$	a, b	S, A, B

$G' = V', T', P', S$

$V' = \{S, A, B\}$

$T' = \{a, b\}$

$P' = \{ S \to a/Bb/aA$

$A \to aB$

$B \to a/Aa \}$

S is start symbol.

Q3) Eliminate useless symbol from

$S \to aA/a/B/C$

$A \to aB/\varepsilon$

$B \to aA$

$C \to cCD$

$D \to abd$

Eliminating ε production

Step 1

ov	nv	productions.
ϕ	A	A → ε
A	A	–

Nullable variable is A

Step 2 :

Given productions	Resulting productions.
S → aA/a/B/C	S → aA/a/B/C
A → aB	A → aB
B → aA	B → aA/a
C → cCD	C → cCD
D → abd	D → abd

Eliminating unit productions :

S → aA/a/cCD
A → aB
B → aA/a
C → cCD
D → abd

Eliminating useless symbols.

Step1:

OV	NV	productions.
q	S, B, D	S → a B → a D → abd
S, B, D	S, B, D, A	A → aB
S, B, D, A	S, B, D, A	S → aA B → aA
S, B, D, A	S, B, D, A	–

Step2:

P'	T'	V'
–	–	S
S → a/aA	a	S, A
A → aB	a	S, A, B
B → a/aA	a	S, A, B

$G' = V', T', P', S$

$V' = \{S, A, B\}$

$T' = \{a\}$

$P' = \{ S \to a/aA,\ A \to aB,\ B \to a/aA \}$

S is Start Symbol.

QA) Simplify the following CFG

$S \rightarrow aSa \mid bSb \mid A$

$A \rightarrow aBb \mid bBa$

$B \rightarrow aB \mid bB \mid \varepsilon$

Eliminating ε production.

Step 1:

OV	nV	productions
ϕ	B	$B \rightarrow \varepsilon$
B	B	–

Nullable Variable is B

Step 2:

Given productions	Resulting productions.
$S \rightarrow aSa \mid bSb \mid A$	$S \rightarrow aSa \mid bSb \mid A$
$A \rightarrow aBb \mid bBa$	$A \rightarrow aBb \mid bBa \mid ab \mid ba$
$B \rightarrow aB \mid bB \mid \varepsilon$	$B \rightarrow aB \mid bB \mid a \mid b$

Eliminating unit productions:

$S \rightarrow aSa \mid bSb \mid aBb \mid bBa \mid ab \mid ba$

$A \rightarrow aBb \mid bBa \mid ab \mid ba$

$B \rightarrow aB \mid bB \mid a \mid b$

Simplification.

Step 1:

OV	NV	productions.
q	S, A, B	S → ab/ba A → ab/ba B → a/b
S, A, B	S, A, B	S → aSa/bSb/aBb/bBa A → aBb/bBa B → aB/bB
S, A, B	S, A, B	-

Step 2:

P'	T'	V'
-	-	S
S → ab/ba/asa/ bsb/aBb/bBa	a, b	S, B
B → a/b/aB/bB	a, b	S, B

$G' = V', T', P', S$

$V' = \{S, B\}$

$T' = \{a, b\}$

$P' = \{S \to ab/ba/asa/bSb/aBb/bBa$
$B \to a/b/aB/bB\}$

S is Start Symbol.

Derivations and Parse Trees :

The grammatical structure of a string is captured by a parse tree, which records which rules were applied to which nonterminals during the string's derivation.

In left_most derivation at each step, the leftmost nonterminal in the working string is choosen for expansion.

In right_most derivation at each step, the right_most nonterminal in the working string is choosen for expansion.

Ambiguity

What is ambiguous grammar ?

Let $G = (V, T, P, S)$ be a CFG. A grammar G is ambiguous if and only if there exists atleast one string $w \in T^*$ for which two or more parse tree exist by applying either the left most derivation or right most derivation.

Q) Consider the grammar shown below from which any arithmetic expression can be obtained.

$E \rightarrow E+E$

$E \rightarrow E-E$

$E \rightarrow E*E$

$E \rightarrow E/E$

$E \rightarrow (E) \mid I$

$E \rightarrow id$

Show the grammar is ambiguous.

Terminals are +, −, *, /, (,), id

String is id +id *id

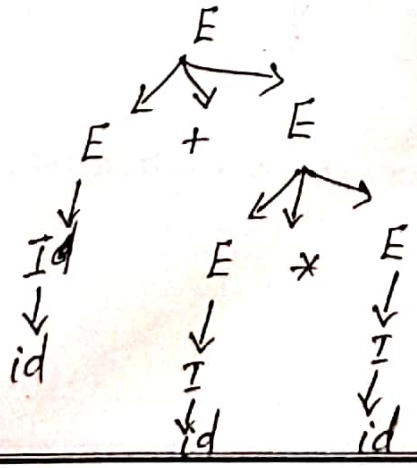
$E \Rightarrow E+E$

$\Rightarrow id+E$

$\Rightarrow id+E*E$

$\Rightarrow id+id*E$

$\Rightarrow id+id*id$



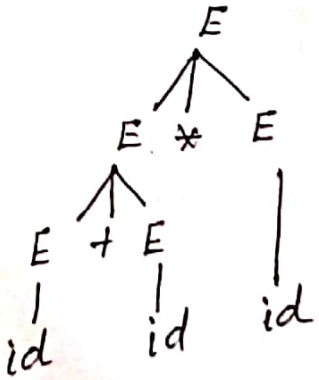
$E \Rightarrow E*E$

$\Rightarrow E+E*E$

$\Rightarrow id+E*E$

$\Rightarrow id+id*E$

$\Rightarrow id+id*id$



Hence it is ambiguous.

(12) Is the following grammar ambiguous.

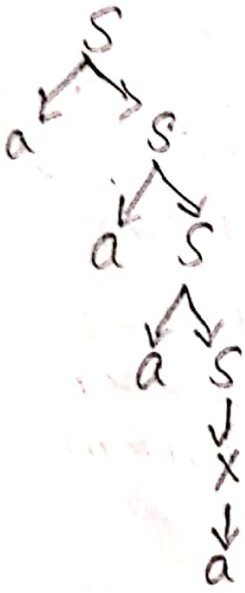
$S \rightarrow aS/X$

$X \rightarrow aX/a.$

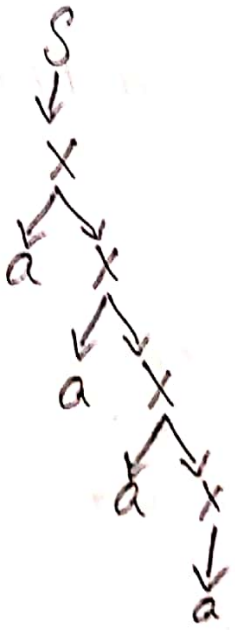
Terminals are a

String is aaaa.

$S \Rightarrow aS$
$\Rightarrow aaS$
$\Rightarrow aaaS$
$\Rightarrow aaaX$
$\Rightarrow aaaa$



$S \Rightarrow X$
$\Rightarrow aX$
$\Rightarrow aaX$
$\Rightarrow aaaX$
$\Rightarrow aaaa$



∴ Hence the grammar is ambiguous.

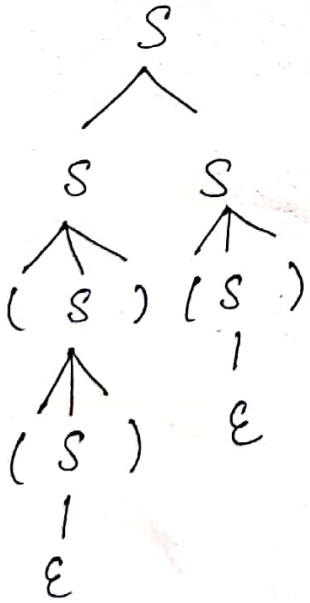
What is inherently ambiguous grammar?

A grammar G is ambiguous if there exists some string $w \in L(G)$ for which there are two or more distinct derivation trees. If there exists a language L for which there is no unambiguous grammar.

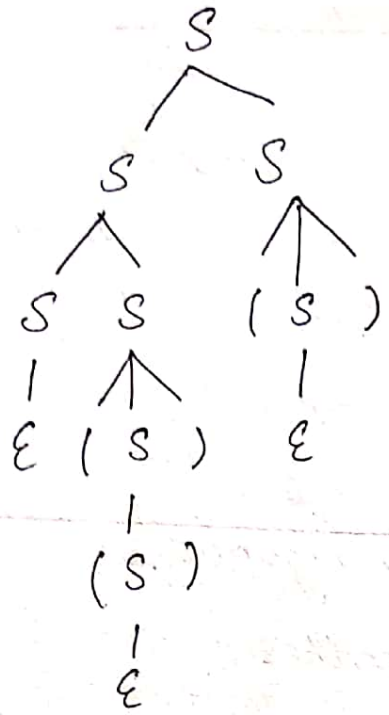
2) Show the given grammar is ambiguous.

Bal = { w ∈ {), (}* : the parentheses are balanced }. Let G = { { S,), (}, {), (}, R, S

$$R = \{ S \to (S) \quad S \to SS \quad S \to \varepsilon \}$$



$S \Rightarrow SS$
$\Rightarrow (S)S$
$\Rightarrow ((S))S$
$\Rightarrow (())S$
$\Rightarrow (())(S)$
$\Rightarrow (())()$



$S \Rightarrow SS$
$\Rightarrow SSS$
$\Rightarrow SS$
$\Rightarrow (S)S$
$\Rightarrow ((S))S$
$\Rightarrow (())S$
$\Rightarrow (())(S)$
$\Rightarrow (())()$

Hence proved.

Techniques for reducing ambiguity

Q) Find the equivalent unambiguous grammar.

Let $G = \{\{S, T, A, B, C, a, b, c\}, \{a, b, c\}, R, S\}$

where $R = \{$
$S \rightarrow aTa$
$T \rightarrow ABC$
$A \rightarrow aA \mid C$
$B \rightarrow Bb \mid C$
$C \rightarrow c \mid \varepsilon$

Eliminate ε productions.

ϕv	nv	productions.
ϕ	C	$C \to \varepsilon$
C	A, B, C	$A \to C$ $B \to C$
A, B, C	T, A, B, C	$T \to ABC$
T, A, B, C	T, A, B, C	–

T, A, B, C are nullable variable.

Given productions	Resulting productions.
$S \to aTa$ $T \to ABC$ $A \to aA \mid C$ $B \to Bb \mid C$ $C \to \in$	$S \to aTa \mid aa$ $T \to ABC \mid BC \mid AC \mid AB \mid C \mid A \mid B$ $A \to aA \mid C \mid a$ $B \to Bb \mid C \mid b$ $C \to \in$

Include atmost one epsilon state

S is not nullable variable.

So no need to include atmost one ε.

Eliminating symmetric recursive rules.

There is no symmetric recursive rule.

Equivalent unambiguous grammar

$G = V, T, P, S$

$V = \{ A, B, C, S, S^*, T \}$

$T = \{ a, b, c, \varepsilon \}$

$P = \{ S^* \to A/B/C/T$

$S^* \to \varepsilon$

$S \to aTa/aa$

$T \to ABC/AB/AC/BC/A/B/C$

$A \to aA/C/a$

$B \to Bb/C/b$

$C \to u$

S^* is start symbol.

---

(2) Find the equivalent unambiguous grammar.

$Bal = \{ w \in \{), (\}^* :$ the parantheses are balanced$\}$. Let $G = \{ \{S,), (\}, \{), (\}, R, S$

$R \Rightarrow \{ S \to (S)$

$S \to SS$

$S \to \varepsilon \}$

Step1: Eliminating ε production.

ov	nv	productions:
ϕ	S	$S \to \varepsilon$
S	S	–

Nullable variable is S

Step2:

Given productions	Resulting productions.
$S \to (S)$	$S \to (S) \mid ()$
$S \to SS$	$S \to SS \mid S$

Include atmost one epislon

$S^* \to S$

$S^* \to \varepsilon$

$S \to (S) \mid () \mid SS$

Eliminate symmetric recursive rules.

$S^* \to S$

$S^* \to \varepsilon$

$S \to SS_1$

$S \to S_1$

$S_1 \to (S) \mid ()$

Equivalent unambiguous grammar is

$G = V, T, P, S$

$V = \{ S^*, S, S_1 \}$

$T = \{ \varepsilon, (,) \}$

$P = \{ S^* \to S$
$S^* \to \varepsilon$
$S \to SS_1$
$S \to S_1$
$S_1 \to (S) \mid () \}$

S^* is start symbol.

For each of the following grammar G show that G is ambiguous. Then find an equivalent grammar that is not ambiguous.

a) $(\{S, a, b\}, \{a, b\}, R, S)$ where $R =$
$\{ S \to \varepsilon, S \to aSa, S \to bSb, S \to aSb,$
$S \to bSa, S \to SS \}$

b) $\{\{S, A, B, T, a, c\}, \{a, c\}, R, S$ where
$R = \{ S \to AB, A \to AA, A \to a, B \to Tc, T \to aT$
$T \to a \}$

c) $(\{S, a, b\}, \{a, b\}, R, S$ where $R =$

$\{ S \to aSb,\ S \to bSa,\ S \to SS,\ S \to \varepsilon \}$

d) $(\{S, a, b\}, \{a, b\}, R, S)$ where $R =$

$\{ S \to aSb,\ S \to aaSb,\ S \to \varepsilon \}$

Solutions :

a) $S \to \varepsilon$

$S \to aSa$

$S \to bSb$

$S \to aSb$

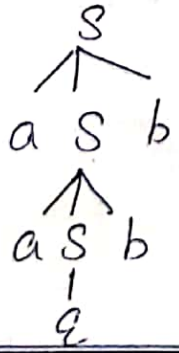
$S \to bSa$

$S \to SS$

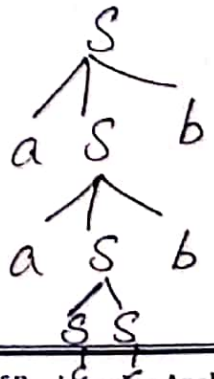
Terminals : $\{\varepsilon, a, b\}$

Let String be : aabb

$S \Rightarrow aSb$
$\Rightarrow aaSbb$
$\Rightarrow aa\varepsilon bb$
$\Rightarrow aabb$



$S \Rightarrow aSb$
$\Rightarrow aaSbb$
$\Rightarrow aaSSbb$
$\Rightarrow aa\varepsilon\varepsilon bb$
$\Rightarrow aabb$



Hence grammar is ambiguous.

Finding equivalent unambiguous grammar

Eliminate ε production

Ov	nV	productions
ϕ	S	S → ε
S	S	S → SS

Nullable variable is S

Given production	Resulting production
S → aSa | bSb | aSb | bSa | SS	S → aSa | bSb | aSb | bSa | SS | aa | bb | ab | ba

Include atmost one epsilon state

S* → S

S* → ε

Eliminate Symmetric recursive rule

S* → S

S* → ε

S1 → aSa | bSb | aSb | bSa | aa | bb | ab | ba

S → SS1

S → S1

$G = V, T, P, S$

$V = \{S, S^*, S_1\}$

$T = \{a, b\}$

$P = \{$ $S^* \to \varepsilon$

$S^* \to S$

$S \to SS_1$

$S \to S_1$

$S_1 \to aSa|bSb|aSb|bSa|aa|bb|ab|ba$

S^* is start symbol.

b) $S \to AB$

$A \to AA$

$A \to a$

$B \to Tc$

$T \to aT$

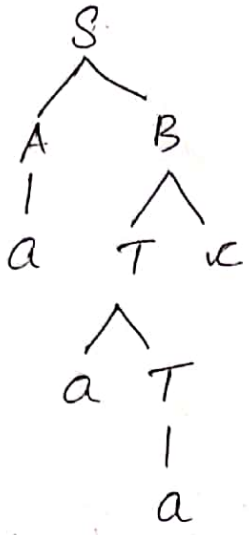
$T \to a$

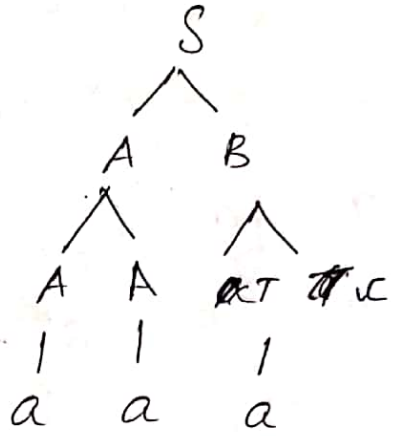
Terminals : $\{c, a\}$

Let string be aaac

$S \Rightarrow AB$
$\Rightarrow aB$
$\Rightarrow aTc$
$\Rightarrow aaTc$
$\Rightarrow aaac$

$S \Rightarrow AB$
$\Rightarrow AAB$
$\Rightarrow aAB$
$\Rightarrow aaB$
$\Rightarrow aaTc$
$\Rightarrow aaac$





Hence proved.

Eliminate Symmetric recursive rule:

$S \rightarrow AB$

$A \rightarrow AA_1$

$A \rightarrow A_1$

$A_1 \rightarrow a$

$B \rightarrow Tc$

$T \rightarrow aT$

$T \rightarrow a$

$G = V, T, P, S$

$V = \{ S, A, A_1, T \}$

$T = \{ a, c \}$

$P = \{ S \rightarrow AB$
$A \rightarrow AA_1$
$A \rightarrow A_1$
$A_1 \rightarrow a$
$B \rightarrow Tc$
$T \rightarrow aT$
$T \rightarrow a \}$

S is start symbol.

c) $S \to aSb$
$S \to bSa$
$S \to SS$
$S \to \varepsilon$

Eliminate ε productions.

Step1:

ov	nv	productions.
q	S	$S \to \varepsilon$
s	S	$S \to SS$

Nullable Variable is S

Given productions	Resulting productions
$S \to aSb \mid bSa \mid SS$	$S \to aSb \mid bSa \mid SS \mid ab \mid ba$

Include atmost one epsilon state.

$S^* \to \varepsilon$
$S^* \to S$

Eliminate symmetric recursive rules:

$S^* \to \varepsilon$
$S^* \to S$
$S \to SS_1$
$S \to S_1$
$S_1 \to aSb \mid bSa \mid ab \mid ba$

$G = V, T, P, S$

$V = \{ S^*, S, S_1 \}$

$T = \{ a, b, \varepsilon \}$

$P = \{ S^* \to \varepsilon$
$S^* \to S$
$S \to SS_1$
$S \to S_1$
$S_1 \to aSb \mid bSa \mid ab \mid ba \}$

S^* is start symbol.

d) $S \to aSb$
$S \to aaSb$
$S \to \varepsilon$

Eliminate epsilon production.

Step1:

OV	nV	production
ϕ	S	$S \to \varepsilon$
S	S	–

Nullable Variable is S

Step2:

Given production	Resulting production
$S \to aSb$ $S \to aaSb$	$S \to aSb \mid ab$ $S \to aaSb \mid aab$

Include almost one epsilon state

$S^* \rightarrow \varepsilon$

$S^* \rightarrow S$

$G = V, T, P, S$

$V = \{S, S^*\}$

$T = \{a, b, \varepsilon\}$

$P = \{ S^* \rightarrow S$
$S^* \rightarrow \varepsilon$
$S \rightarrow aSb | aaSb | ab | aab \}$

S^* is start symbol.

Q) Reduce the ambiguity from the given grammar $G = \{\{E, id, *, +, (,), \}, \{id, +, *, (,)\}, R, E$ where $R =$

$E \rightarrow E+E$

$E \rightarrow E*E$

$E \rightarrow (E)$

$E \rightarrow id$

Eliminate symmetric recursive rules.

$E \rightarrow E + T$

$E \rightarrow T$

$T \rightarrow T * T$

$T \rightarrow (E)$

$T \rightarrow (id)$

$T \rightarrow T * F$

$T \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow id$

$G = V, T, P, S$

$V = \{ E, T, F \}$

$T = \{ +, *, (,), id \}$

$P = \quad E \rightarrow E + T$

$E \rightarrow T.$

$T \rightarrow T * F$

$T \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow id \quad \}$

E is start symbol.

CNF: Chomsky Normal Form

The grammar G is said to be in CNF if all productions are of the form $A \rightarrow BC$

or

$A \rightarrow a$

Q1) Obtain grammar in CNF

$S \rightarrow 0A \mid 1B$

$A \rightarrow 0AA \mid 1S \mid 1$

$B \rightarrow 1BB \mid 0S \mid 0$

Given productions	Resulting productions.
$S \rightarrow 0A \mid 1B$	$S \rightarrow B_0A \mid B_1B$ $B_0 \rightarrow 0$ $B_1 \rightarrow 1$
$A \rightarrow 0AA \mid 1S \mid 1$	$A \rightarrow B_0AA \mid B_1S \mid 1$ $B_0 \rightarrow 0$ $B_1 \rightarrow 1$
$B \rightarrow 1BB \mid 0S \mid 0$	$B \rightarrow B_1BB \mid B_0S \mid 0$ $B_1 \rightarrow 1$ $B_0 \rightarrow 0$

Step2:

Given productions	Resulting productions.
$S \rightarrow B_0A \mid B_1B$	$S \rightarrow B_0A \mid B_1B$ $B_0 \rightarrow 0$ $B_1 \rightarrow 1$
$A \rightarrow B_0AA \mid B_1S \mid 1$	$A \rightarrow B_0D_1 \mid B_1S \mid 1$ $D_1 \rightarrow AA$
$B \rightarrow B_1BB \mid B_0S \mid 0$	$B \rightarrow B_1D_2 \mid B_0S \mid 0$ $D_2 \rightarrow BB$

$G = V, T, P, S$

$V = \{ S, A, B, B_0, B_1, D_1, D_2 \}$

$T = \{0, 1\}$

$P = \{ S \to B_0 A \mid B_1 B$

$A \to B_0 D_1 \mid B_1 S \mid 1$

$B \to B_1 D_2 \mid B_0 S \mid 0$

$B_0 \to 0$

$B_1 \to 1$

$D_1 \to AA$

$D_2 \to BB \}$

S is Start symbol.

---

b) Obtain the grammar in CNF. Let $G = (\{S, A, B, C, a, u\}, \{A, B, C\}, R, S)$

$S \to aACa$

$A \to B \mid a$

$B \to C \mid u$

$C \to uC \mid \varepsilon$

Eliminate ε productions.

Step 1:

OV	NV	productions.
ϕ	C	$C \to \varepsilon$
C	B, C	$B \to C$
B, C	A, B, C	$A \to B$
A, B, C	A, B, C	–

Nullable variables are A, B, C

Step2:

Given productions	Resulting productions.
$S \to aACa$ $A \to B \mid a$ $B \to C \mid \epsilon$ $C \to \epsilon C$	$S \to aACa \mid aCa \mid aAa \mid aa$ $A \to a \mid B$ $B \to C \mid \epsilon$ $C \to \epsilon C \mid \epsilon$

Eliminate unit productions.

$S \to aACa \mid aCa \mid aAa \mid aa$
$A \to a \mid \epsilon C \mid \epsilon$
$B \to \epsilon C \mid \epsilon$
$C \to \epsilon C \mid \epsilon$

Eliminate useless symbols

$S \to aACa \mid aCa \mid aAa \mid aa$
$A \to a \mid \epsilon C \mid \epsilon$
$C \to \epsilon C \mid \epsilon$

Obtain grammar in CNF

Step1:

Given productions.	Resulting productions.
$S \to aACa \mid aCa \mid aAa \mid aa$	$S \to B_a ACB_a \mid B_a CB_a \mid B_a ABa \mid B_a B_a$ $B_a \to a$
$A \to a \mid \epsilon C \mid \epsilon$	$A \to a \mid B_c C \mid \epsilon$ $B_c \to c$
[illegible]	[illegible]
$C \to \epsilon C \mid \epsilon$	$C \to B_c C \mid \epsilon$ $B_c \to c$

Step 2:

Given productions	Resulting productions.
$S \to BaACBa \mid BaCBa \mid BaABa \mid BaBa$	$S \to D_2 D_1 \mid Ba D_1 \mid BaD_3 \mid BaBa$ $D_2 \to BaA$ $D_1 \to CBa$ $D_3 \to ABa$ $Ba \to a$
$A \to a \mid Bc C \mid \in$	$A \to a \mid Bc C \mid \in$ $Bc \to \in$
[illegible]	[illegible]
$C \to Bc C \mid \in$	$C \to Bc C \mid \in$

Complete $G = V, T, P, S$

Q3) Obtain the grammar in CNF.

$S \to aSa$
$S \to B$
$B \to bbC$
$B \to bb$
$C \to \in$
$C \to \in C$

Eliminate ε productions:

Step1:

ov	nv	productions
ϕ	C	C → ε
C	.C	–

Nullable Variable is C

Step2:

Given productions	Resulting productions
S → aSa/B	S → aSa/B
B → bbC/bb	B → bbC/bb
C → xC	C → xC/x

Eliminate unit productions:

S → aSa/bbC/bb
B → bbC/bb
C → xC/x

Eliminate useless symbols

S → aSa/bbC/bb
C → xC/x

Obtain grammar in CNF

Step1:

Given productions	Resulting productions.
S → aSa/bbC/bb	S → BaSBa/BbBbC/BbBb
B → bbC/bb	~~B → BBC/BB~~
C → xC/x	C → BcC/x
	Ba → a, Bb → b, Bc → x

Step 2:

Given productions	Resulting productions.
$S \to B_a S B_a / B_b B_b C / B_b B_b$	$S \to B_a D_1 / B_b D_2 / B_b B_b$ $D_1 \to S B_a$ $D_2 \to B_b C$
$B \to B_b B_b C / B_b B_b$	$B \to B_b D_2 / B_b B_b$
$C \to B_\epsilon C / \epsilon$	$C \to B_\epsilon C / \epsilon$ $B_a \to a$ $B_b \to b$ $B_\epsilon \to \epsilon$

Complete $G = V, T, P, S$

Q4) Obtain grammar in CNF

$S \to ABC$

$A \to aC / D$

$B \to bB / \epsilon / A$

$C \to AC / \epsilon / C\epsilon$

$D \to aa$

Eliminate ε productions.

Step1:

ov	nv	productions.
ϕ	B, C	B → ε C → ε
B, C	B, C	-

Nullable variables are B, C

Step2:

Given productions.	Resulting productions.
S → ABC	S → ABC | AC | AB | A
A → aC | D	A → aC | D | a
B → bB | A	B → bB | A | b
C → AC | Cc	C → AC | Cc | A | c
D → aa	D → aa

Eliminate unit productions.

S → ABC | AC | AB | aC | aa | a
A → aC | aa | a
B → bB | aC | aa | a | b
C → AC | Cc | aC | aa | a | c
D → aa

Eliminate useless productions.

S → ABC | AC | AB | aC | aa | a
A → aC | aa | a
B → bB | aC | aa | a | b
C → AC | Cc | aC | aa | a | c

Obtain grammar in CNF

Step1:

Given productions	Resulting productions
$S \to ABC \mid AC \mid AB \mid aC \mid aa \mid a$	$S \to ABC \mid AC \mid AB \mid BaC \mid BaBa \mid a$
$A \to aC \mid aa \mid a$	$A \to BaC \mid BaBa \mid a$ $Ba \to a$
$B \to bB \mid aC \mid aa \mid a \mid b$	$B \to BbB \mid BaC \mid BaBa \mid a \mid b$
$C \to AC \mid Cc \mid aC \mid aa \mid a \mid \epsilon$	$C \to AC \mid CBc \mid BaC \mid BaBa \mid a \mid \epsilon$

Step2:

Given productions	Resulting productions
$S \to ABC \mid AC \mid AB \mid BaC \mid BaBa \mid a$	$S \to AD_1 \mid AC \mid AB \mid BaC \mid BaBa \mid a$ $D_1 \to BC$
$A \to BaC \mid BaBa \mid a$	$A \to BaC \mid BaBa \mid a$ $Ba \to a$
$B \to BbB \mid BaC \mid BaBa \mid a \mid b$	$B \to BbB \mid BaC \mid BaBa \mid a \mid b$
$C \to AC \mid CBc \mid BaC \mid BaBa \mid a \mid \epsilon$	$C \to AC \mid CBc \mid BaC \mid BaBa \mid a \mid \epsilon$

Complete $G = V, T, P, S$

Q5) Obtain grammar in CNF

$S \rightarrow aTVa$

$T \rightarrow aTa \mid bTb \mid \varepsilon \mid V$

$V \rightarrow cVc \mid \varepsilon$

Eliminate ε production

Step1:

OV	nV	productions
ϕ	T, V	$T \rightarrow \varepsilon$ $V \rightarrow \varepsilon$
T, V	T, V	$T \rightarrow V$
TV	T, V	–

Nullable Variables are T, V

Step2:

Given productions	Resulting productions
$S \rightarrow aTVa$ $T \rightarrow aTa \mid bTb \mid V$ $V \rightarrow cVc$	$S \rightarrow aTVa \mid aVa \mid aTa \mid aa$ $T \rightarrow aTa \mid bTb \mid V \mid aa \mid bb$ $V \rightarrow cVc \mid cc$

Eliminate unit productions.

$S \rightarrow aTVa \mid aVa \mid aTa \mid aa$

$T \rightarrow aTa \mid bTb \mid cVc \mid cc \mid aa \mid bb$

$V \rightarrow cVc \mid cc$

Obtain grammar in CNF

Step1:

Given productions	Resulting productions.
$S \to aTVa \mid aVa \mid aTa \mid aa$	$S \to BaTVBa \mid BaVBa \mid BaTBa \mid BaBa$ $Ba \to a$
$T \to aTa \mid bTb \mid cVc \mid cXc \mid aa \mid bb$	$T \to BaTBa \mid BbTBb \mid BcVBc \mid BcBc \mid BaBa \mid BbBb$ $Bb \to b$ $Ba \to a$ $Bc \to c$
$V \to cVc \mid cXc$	$V \to BcVBc \mid BcBc$ $Bc \to c$

Step2:

Given productions	Resulting productions.
$S \to BaTVBa \mid BaVBa \mid BaTBa \mid BaBa$	$S \to D_2D_1 \mid BaD_1 \mid BaD_3 \mid BaBa$ $D_2 \to BaT$ $D_1 \to VBa$ $D_3 \to TBa$
$T \to BaTBa \mid BbTBb \mid BcVBc \mid BcBc \mid BaBa \mid BbBb$	$T \to BaD_3 \mid BbD_4 \mid BcD_5 \mid BcBc \mid BaBa \mid BbBb$ $D_4 \to TBb$ $D_5 \to VBc$

$V \to B_c V B_c \mid B_c B_c$

$V \to B_c D_5 \mid B_c B_c$

$D_5 \to V B_c$

$B_a \to a$

$B_b \to b$

$B_c \to c$

$G = V, T, P, S$

$V = \{ S, T, V, B_a, B_b, B_c, D_1, D_2, D_3, D_4, D_5 \}$

$T = \{ a, b, c \}$

$P = \{ S \to D_2 D_1 \mid B_a D_1 \mid B_a D_3 \mid B_a B_a$

$D_2 \to B_a T$

$D_1 \to V B_a$

$D_3 \to T B_a$

$T \to B_a D_3 \mid B_b D_4 \mid B_c D_5 \mid B_c B_c \mid B_a B_a \mid B_b B_b$

$D_4 \to T B_b$

$D_5 \to V B_c$

$V \to B_c D_5 \mid B_c B_c$

$B_a \to a$

$B_b \to b$

$B_c \to c \}$

S is start symbol

Steps:

Q) Simplify the CFG

or

Eliminate useless symbols / productions.

Step1: Eliminate ε production

Step2: Eliminate unit production

Step3: Simplify.

Q) Reduce the ambiguity from grammar

or

Find an equivalent unambiguous grammar

Step1: Eliminate ε production

- Include atmost one epsilon state

Step2: Eliminate Symmetric recursive rules.

Q) To Obtain grammar in CNF

Step1: Eliminate ε production

Step2: Eliminate unit production

Step3: Apply CNF rules.

Pushdown Automata

A pushdown Automata is a seven tuple machine where $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

Q is set of finite states
Σ is set of i/p alphabets
Γ is set of stack alphabets.
δ is transition from $Q \times (\Sigma \cup \epsilon) \times \Gamma$ to finite subset of $Q \times \Gamma$
q_0 is initial state
Z_0 is initial symbol on the stack
F is set of final states.

The actions (ie transitions) performed by PDA depends on

1. Current state
2. Next i/p symbol
3. The symbol on top of the stack.

The action performed by the machine consists of.

1. Changing the states from one state to another.
2. Replacing the symbol on the stack.

Construction of PDA

Q1) Obtain a PDA to accept the language $L(M) = \{WCW^R / W \in (a+b)^*\}$ where W^R is reverse of W by a final state.

$W = abb$

$W^R = bba$

$WCW^R = abbCbba$

which is a string of palindrome.

$\delta(q_0, a, Z_0) = q_0, aZ_0$

$\delta(q_0, a, a) = q_0, aa$

$\delta(q_0, b, a) = q_0, ba$

$\delta(q_0, b, Z_0) = q_0, bZ_0$

$\delta(q_0, a, b) = q_0, ab$

$\delta(q_0, b, b) = q_0, bb$

$\delta(q_0, C, Z_0) = q_1, Z_0$

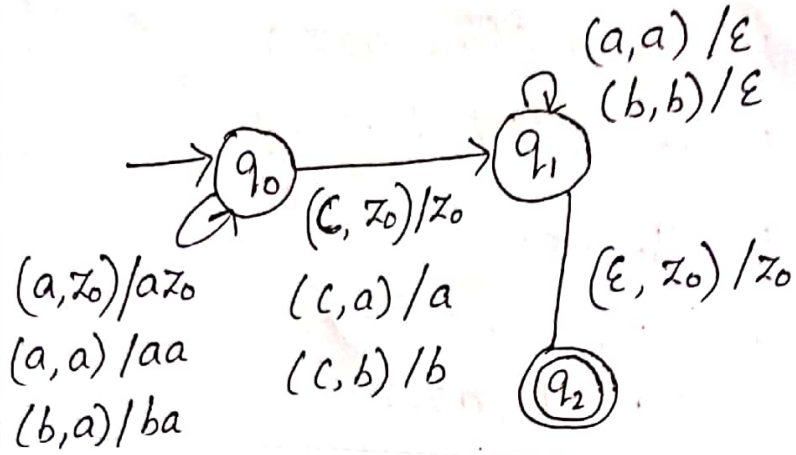
$\delta(q_0, C, a) = q_1, a$

$\delta(q_0, C, b) = q_1, b$

$\delta(q_1, a, a) = q_1, \varepsilon$

$\delta(q_1, b, b) = q_1, \varepsilon$

$\delta(q_1, \epsilon, Z_0) = q_2, Z_0$



To accept the string:

$(q_0, aabCbaa, Z_0) \vdash (q_0, abCbaa, aZ_0)$
$\vdash (q_0, bCbaa, aaZ_0)$
$\vdash (q_0, Cbaa, baaZ_0)$
$\vdash (q_1, baa, baaZ_0)$
$\vdash (q_1, aa, aaZ_0)$
$\vdash (q_1, a, aZ_0)$
$\vdash (q_1, \epsilon, Z_0)$
$\vdash (q_2, Z_0)$

To reject the string:

$(q_0, aabCbab, Z_0) \vdash (q_0, abCbab, aZ_0)$
$\vdash (q_0, bCbab, aaZ_0)$
$\vdash (q_0, Cbab, baaZ_0)$
$\vdash (q_1, bab, baaZ_0)$
$\vdash (q_1, ab, aaZ_0)$
$\vdash (q_1, b, aZ_0)$

Q2) Obtain a PDA to accept the language $L = \{a^n b^n / n \geq 1\}$ by a final state.

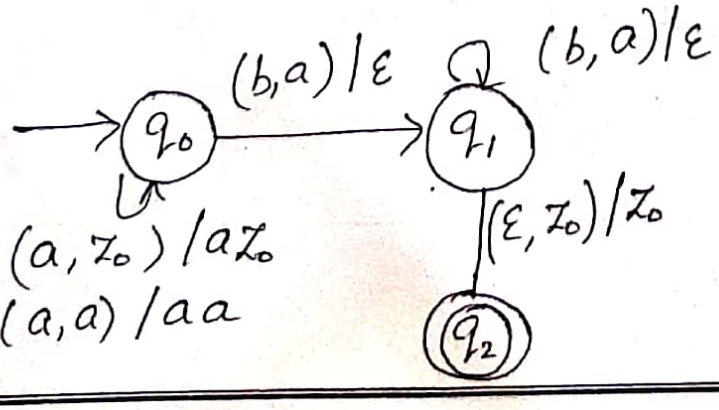
$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, \varepsilon)$

$\delta(q_1, b, a) = (q_1, \varepsilon)$

$\delta(q_1, \varepsilon, Z_0) = (q_2, Z_0)$



Q3) Obtain a PDA for the language $L = \{ w \in \{a,b\}^* : \#_a(w) = \#_b(w) \}$

$\delta(q_0, a, Z_0) = q_0, aZ_0$

$\delta(q_0, a, a) = q_0, aa$

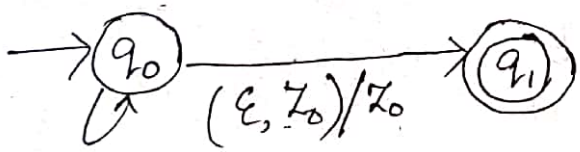
$\delta(q_0, b, a) = q_0, \varepsilon$

$\delta(q_0, b, Z_0) = q_0, bZ_0$

$\delta(q_0, a, b) = q_0, \varepsilon$

$\delta(q_0, b, b) = q_0, bb$

$\delta(q_0, \varepsilon, Z_0) = q_1, Z_0$



$(a, Z_0)/aZ_0$

$(a, a)/aa$

$(b, a)/\varepsilon$

$(b, Z_0)/bZ_0$

$(a, b)/\varepsilon$

$(b, b)/bb$

Q4) Obtain a non deterministic PDA for

$PalEven = \{ WW^R : W \in \{a,b\}^* \}$

$W = abb$

$W^R = bba$

$WW^R = abbbba$

$\delta(q_0, a, Z_0) = q_0, aZ_0$

$\delta(q_0, a, a) = q_0, aa$

$\delta(q_0, b, a) = q_0, ba$

$\delta(q_0, b, Z_0) = q_0, bZ_0$

$\delta(q_0, b, b) = q_0, bb$

$\delta(q_0, a, b) = q_0, ab$

$\delta(q_0, \varepsilon, Z_0) = q_1, Z_0$

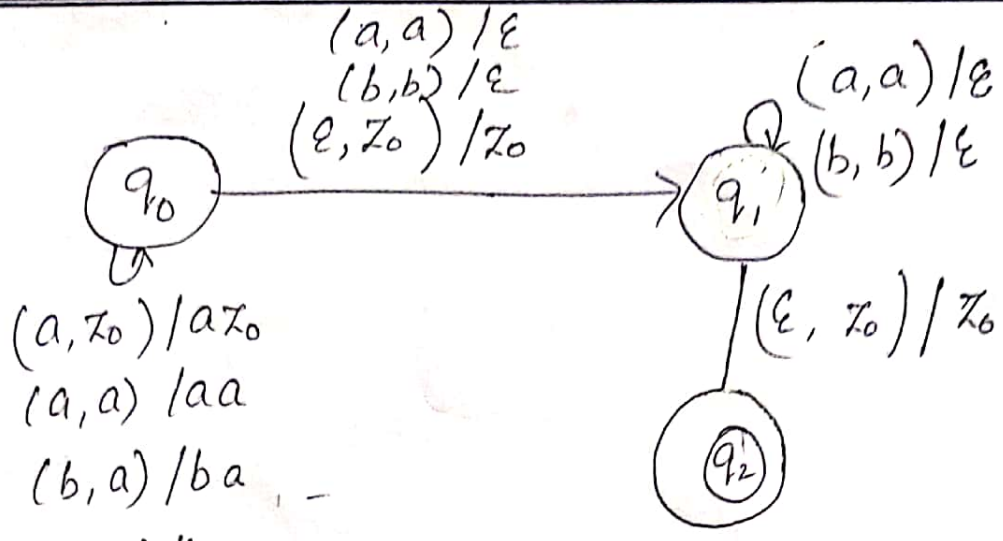
Assume that W^R begins.

$\delta(q_0, a, a) = q_1, \varepsilon$

$\delta(q_0, b, b) = q_1, \varepsilon$

$\delta(q_1, a, a) = q_1, \varepsilon$

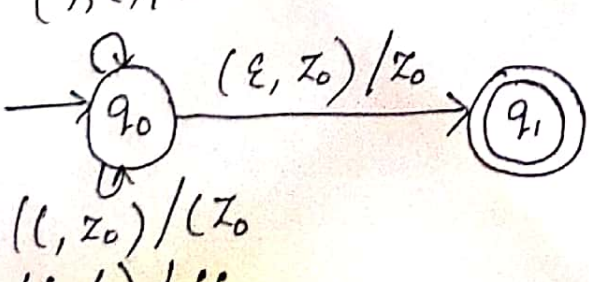
$\delta(q_1, bb) = q_1, \varepsilon$



(a, Z0)/aZ0
(a, a)/aa
(b, a)/ba
(b, Z0)/bZ0
(b, b)/bb
(a, b)/ab

Q5) Consider Bal = {w ∈ {), (}* : the parantheses are balanced}

δ(q0, (, Z0) = q0, (Z0
δ(q0, (, () = q0, ((
δ(q0,), () = q0, ε
δ(q0, ε, Z0) = q1, Z0



Q6) Construct a PDA for $L = \{a^n b^{2n} : n \geq 0\}$

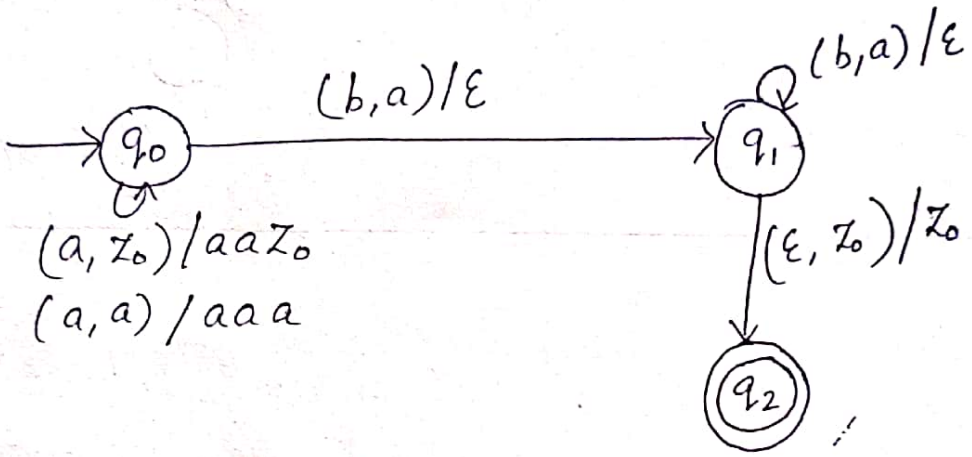
$\delta(q_0, a, Z_0) = q_0, aaZ_0$

$\delta(q_0, a, a) = q_0, aaa$

$\delta(q_0, b, a) = q_1, \varepsilon$

$\delta(q_1, b, a) = q_1, \varepsilon$

$\delta(q_1, \varepsilon, Z_0) = q_2, Z_0$



Q7) Construct a PDA for $L = \{a^m b^n : m \neq n;\ m, n > 0\}$

$\delta(q_0, a, Z_0) = q_0, aZ_0$

$\delta(q_0, a, a) = q_0, aa$

$\delta(q_0, b, a) = q_1, \varepsilon$

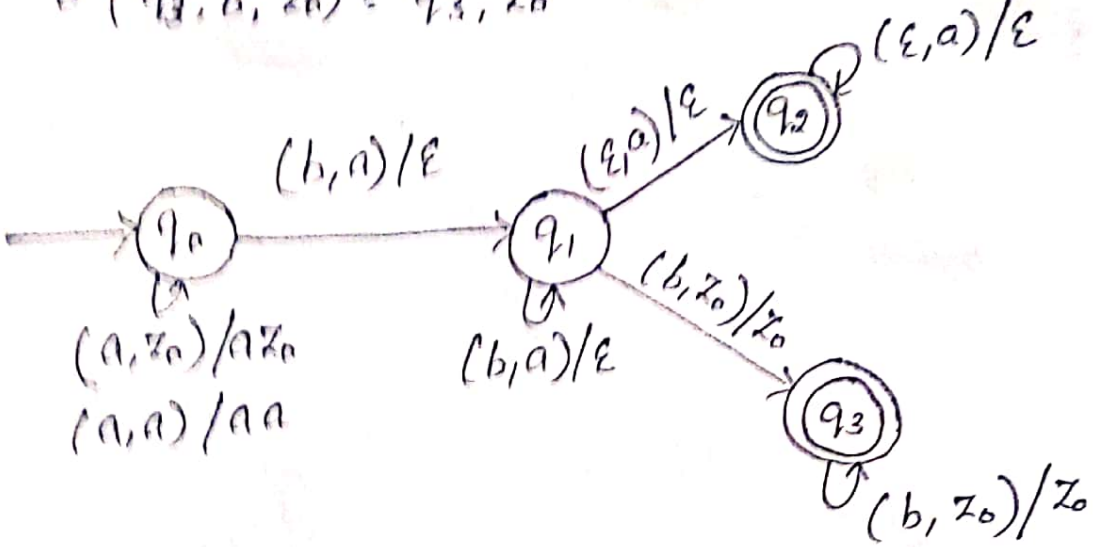
$\delta(q_1, b, a) = q_1, \varepsilon$

$\delta(q_1, \varepsilon, a) = q_2, \varepsilon$

$\delta(q_2, \varepsilon, a) = q_2, \varepsilon$

$\delta(q_1, b, Z_0) = q_3, Z_0$

$\delta(q_3, b, Z_0) = q_3, Z_0$



Note:

Let $A^nB^nC^n = \{a^nb^nc^n : n \geq 0\}$. Obtain a PDA.

No PDA exists to accept this language.

Let $L = \neg A^nB^nC^n$

There is a PDA to accept L

$L = L1 \cup L2$ where

$L1 = \{w \in \{a,b,c\}^* :$ the letters are out of order$\}$

$L2 = \{a^ib^jc^k : i,j,k \geq 0$ and $(i \neq j$ or $j \neq k\}$

(in other words not equal numbers of a's, b's and c's)

Deterministic and Non-deterministic PDA

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA.
The PDA is deterministic if

1. $\delta(q, a, Z)$ has only one element.
2. If $\delta(q, \varepsilon, Z)$ is not empty, then $\delta(q, a, Z)$ should be empty.

Techniques for reducing Nondeterminism

In the given language

$L = \{a^m b^n : m \neq n, m, n > 0\}$

We saw nondeterminism arising from two very specific circumstances:

*) A transition that should be taken only if the stack is empty competes against one or more moves that require a match of some string on the stack

*) A transition that should be taken only if the input stream is empty competes against one or more moves that require a match against a specific input character.

Using a bottom of stack marker.

$\delta(q_1, b, \#) = q_3, \varepsilon$

$\delta(q_2, \varepsilon, \#) = q_2, \varepsilon$

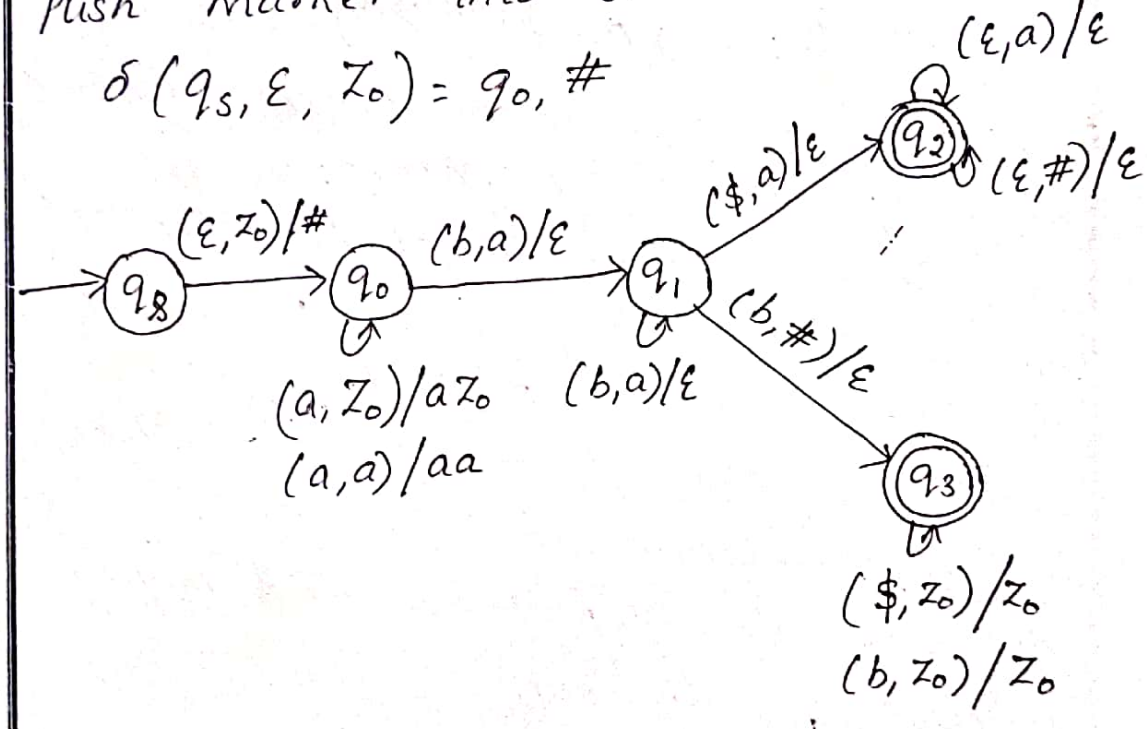
Using an End-of-String marker

$\delta(q_1, \$, a) = q_2, \varepsilon$

$\delta(q_3, \$, Z_0) = q_3, Z_0$

Push marker into stack

$\delta(q_s, \varepsilon, Z_0) = q_0, \#$



Equivalence of Context-Free Grammars & PDAs

Building a PDA from a grammar

Topdown parsing

Algorithm :

- Start up transition

  $\delta(P, \varepsilon, \varepsilon) = q, S$

- For each rule $X \rightarrow \gamma_1, \gamma_2 \ldots \gamma_n$

  Transition : $\delta(q, \varepsilon, X) = q, \varepsilon$

- For each character $c \in \Sigma$

  Transition : $\delta(q, c, c) = q, \varepsilon$

Bottom up parsing

Algorithm

- Shift transition : $\delta(P, c, \varepsilon) = P, c$
- Reduce transition

  $\delta(P, \varepsilon, (\gamma_1, \gamma_2 \ldots \gamma_n)^R) = P, X$

- Finish up transition

  $\delta(P, \varepsilon, S) = q, \varepsilon$

a) Use cfgtoPDAtopdown and cfgtoPDAbottomup to convert given CFG to PDA

R = $E \to E+T$

$E \to T$

$T \to T*F$

$T \to F$

$F \to (E)$

$F \to id$

Topdown parsing

$\delta(q_0, \varepsilon, Z_0) = q_1, EZ_0$

$\delta(q_1, \varepsilon, E) = q_1, E+T$

$\delta(q_1, \varepsilon, E) = q_1, T$

$\delta(q_1, \varepsilon, T) = q_1, T*F$

$\delta(q_1, \varepsilon, T) = q_1, F$

$\delta(q_1, \varepsilon, F) = q_1, (E)$

$\delta(q_1, \varepsilon, F) = q_1, id$

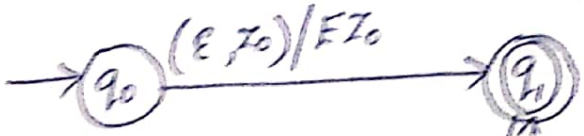
$\delta(q_1, id, id) = q_1, \varepsilon$

$\delta(q_1, (, () = q_1, \varepsilon$

$\delta(q_1,),)) = q_1, \varepsilon$

$\delta(q_1, *, *) = q_1, \varepsilon$

$\delta(q_1, +, +) = q_1, \varepsilon$



$(\varepsilon, E)/E+T$
$(\varepsilon, E)/T$
$(\varepsilon, T)/T*F$
$(\varepsilon, T)/F$
$(\varepsilon, F)/(E)$
$(\varepsilon, F)/id$
$(id, id)/\varepsilon$
$((, ()/\varepsilon$
$(),))/\varepsilon$
$(*, *)/\varepsilon$
$(+, +)/\varepsilon$

Bottom Up parsing:

$\delta(q_0, id, \varepsilon) = q_0, id$

$\delta(q_0, (, \varepsilon) = q_0, ($

$\delta(q_0,), \varepsilon) = q_0,)$

$\delta(q_0, +, \varepsilon) = q_0, +$

$\delta(q_0, *, \varepsilon) = q_0, *$

$\delta(q_0, \varepsilon, T+E) = q_0, E$

$\delta(q_0, \varepsilon, T) = q_0, E$

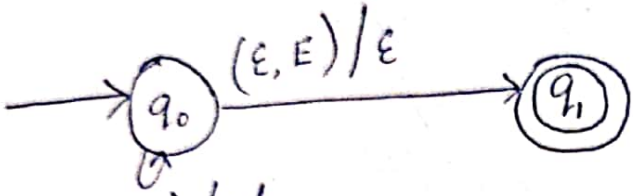
$\delta(q_0, \varepsilon, F*T) = q_0, T$

$\delta(q_0, \varepsilon, F) = q_0, T$

$\delta(q_0, \varepsilon,)E(= q_0, F$

$\delta(q_0, \varepsilon, id) = q_0, F$

$\delta(q_0, \varepsilon, E) = q_1, \varepsilon$



$(id, \varepsilon)/id$

$(, \varepsilon)/($

$), \varepsilon)/)$

$(+, \varepsilon)/+$

$(*, \varepsilon)/*$

$(\varepsilon, T+E)/E$

$(\varepsilon, T)/E$

$(\varepsilon, F*T)/T$

$(\varepsilon, F)/T$

$(\varepsilon,)E()/F$

$(\varepsilon, id)/F$

Building a Grammar from a PDA

Conversion of PDA to CFG

Let $WCW^R = \{wcw^R : w \in \{a,b\}^*\}$

WCW^R

$\delta(q_0, a, Z_0) = q_0, aZ_0$

$\delta(q_0, a, a) = q_0, aa$

$\delta(q_0, b, a) = q_0, ba$

$\delta(q_0, b, Z_0) = q_0, bZ_0$

$\delta(q_0, b, b) = q_0, bb$

$\delta(q_0, a, b) = q_0, ab$

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

$\delta(q_0, c, a) = q_1, a$

$\delta(q_0, c, b) = q_1, b$

$\delta(q_1, a, a) = q_1, \varepsilon$

$\delta(q_1, b, b) = q_1, \varepsilon$

$\delta(q_1, \varepsilon, Z_0) = q_2, Z_0$

Include : $\delta(q_s, \varepsilon, Z_0) = q_0, \#$

Replace Z_0 with Bottom up stack marker

$\delta(q_0, a, \#) = q_0, a\#$

$\delta(q_0, a, a) = q_0, aa$

$\delta(q_0, b, a) = q_0, ba$

$\delta(q_0, b, \#) = q_0, b\#$

$\delta(q_0, b, b) = q_0, bb$

$\delta(q_0, a, b) = q_0, ab$

$\delta(q_0, c, \#) = q_1, \#$

$\delta(q_0, c, a) = q_1, a$

$\delta(q_0, c, b) = q_1, b$

$\delta(q_1, \varepsilon, \#) = q_2, \varepsilon$

Productions:

$S \rightarrow S, \#, a$

$\delta(q_0, a, \#) = q_0, a\#$

$q_0\#q_0 \rightarrow a(q_0aq_0)(q_0\#q_0)$
$\mid a(q_0aq_1)(q_1\#q_0)\mid$
$a(q_0aq_2)(q_2\#q_0)$

$q_0\#q_1 \rightarrow a(q_0aq_0)(q_0\#q_1)\mid$
$a(q_0aq_1)(q_1\#q_1)\mid$
$a(q_0aq_2)(q_2\#q_1)$

$q_0\#q_2 \rightarrow a(q_0aq_0)(q_0\#q_2)\mid$
$a(q_0aq_1)(q_1\#q_2)\mid$
$a(q_0aq_2)(q_2\#q_2)$

$\delta(q_0, a, a) = q_0, aa$	$q_0aq_1 \rightarrow a(q_0aq_1)(q_1aq_1)$
$\delta(q_0, a, b) = q_0, ab$	$q_0bq_1 \rightarrow a(q_0aq_1)(q_1bq_1)$
$\delta(q_0, b, \#) = q_0, b\#$	$q_0\#q_1 \rightarrow b(q_0bq_1)(q_1\#q_1)$
$\delta(q_0, b, b) = q_0, bb$	$q_0bq_1 \rightarrow b(q_0bq_1)(q_1bq_1)$
$\delta(q_0, b, a) = q_0, ba$	$q_0aq_1 \rightarrow b(q_0bq_1)(q_1aq_1)$
$\delta(q_0, C, \#) = q_1, \#$	$q_0\#q_1 \rightarrow C(q_1\#q_1)$
$\delta(q_0, C, a) = q_1, a$	$q_0aq_1 \rightarrow C(q_1aq_1)$
$\delta(q_0, C, b) = q_1, b$	$q_0bq_1 \rightarrow C(q_1bq_1)$
$\delta(q_1, a, a) = q_1, \epsilon$	$q_1aq_1 \rightarrow a(q_1\epsilon q_1)$
$\delta(q_1, b, b) = q_1, \varepsilon$	$q_1bq_1 \rightarrow b(q_1\epsilon q_1)$
$\delta(q_1, \varepsilon, \#) = q_2, \varepsilon$	$q_1\#q_2 \rightarrow \varepsilon(q_2\epsilon q_2)$

$q_0\varepsilon q_0 \rightarrow \varepsilon$

$q_1\epsilon q_1 \rightarrow \varepsilon$

$q_2\epsilon q_2 \rightarrow \varepsilon$

Sharonrojipriya.cse@sairamce.edu.in