Module 3

CONTEXT FREE GRAMMER

Context Free Grammer (CFG):

$G = (V, T, P, S)$

The context Free Grammer is defined by quadruple or 4-tuple where $G = (V, T, P, S)$

where

V - set of variables or non terminals
T - set of terminals
P - set of productions or rules and
S - start variable

All productions are of the form $A \rightarrow \alpha$ where $\alpha \in (V \cup T)^*$ & A is non Terminal

The languages of this CFG are called context free languages (CFL) which are accepted by push down automata (PDA)

1. Write CFG to generate string consisting of any no of a's
$\Sigma = \{a\}$

String is a^*

$G = (V, T, P, S)$

$V = \{S\}$ $\quad P: S \rightarrow aS$

$T = \{a\}$ $\quad S \rightarrow \epsilon$

S = Start Variable

$w = aaaa$

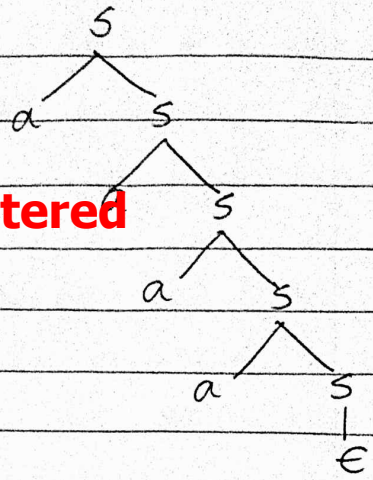
$S \to aS$

$\to aaS$

$\to aaaS$

$\to aaaaS$

$\to aaaa$



2. Strings with a's & b's $\Sigma = \{a, b\}$

$L = \{ab, aa \ldots\}$

$(a+b)^*$

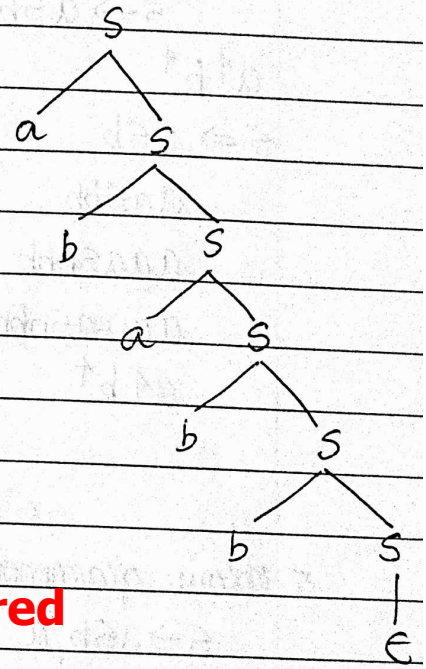
$G = (V, T, P, S)$

$V = \{S$

$T = \{a, b\}$

S = start variable

$S \to aS/bS/\epsilon$



$w = ababb$

$S \to aS$

$\to abS$

$\to abaS$

$\to ababS$

$\to ababbS$

$\to ababb$

3. Obtain CFG to generate even number of a's $\Sigma = \{a\}$

$L = \{aa, aaaa, \ldots\}$

$(aa)^*$

$S \to aaS/\epsilon$

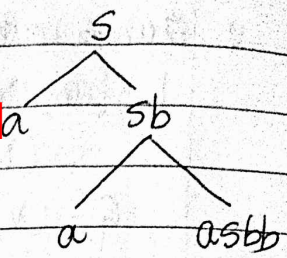
4 Obtain CFG for the language $L=\{a^n b^n \mid n \geq 0\}$

$L=\{ab, aabb, \ldots\}$

$G=(V, T, P, S)$

$T=\{a,b\}$

$S \rightarrow aSb \mid \epsilon$



$a^4 b^4$

$S \Rightarrow aSb$

$aaSbb$

$aaaSbbb$

$aaaaSbbbb$

$a^4 b^4$

5 Obtain grammar for $L=\{a^{n+1} b^n \mid n \geq 0\}$

$S \rightarrow aSb \mid a$

$a^3 b^2$

$S \rightarrow aSb$

$aaSbb$

$aaabb$

$L=\{a^n b^{n+1} \mid n \geq 0\}$

$S \rightarrow aSb \mid b$

Date: YOUVA

6 Obtain grammer $L=\{a^n b^{2n} : n \geq 0\}$

$S \rightarrow aSbb/\epsilon$

a^2b^4

$S \rightarrow aSbb$

$\rightarrow aaSbbbb$

$\rightarrow aabbbb$

4/12/21

7. Write grammer to generate set of palindromes over $\Sigma=\{a,b\}$

$L=\{aa, bb, aba, bab, abba, baab, \ldots\}$

$G=(V,T,P,S) \quad T=\{a,b\}$

$S \rightarrow aSa/bSb/a/b/\epsilon$

$L = abba$

$S \rightarrow aSa$

$abSba \quad (S \rightarrow bSb)$

$abba \quad (S \rightarrow \epsilon)$

8. Obtain CFG for the language given below

$L=\{ww^R / w \in \{a,b\}^*\}$

$L=\{abba, baab, abbbba, \ldots\}$

All strings of ww^R are palindrome but all palindrome are not ww^R

$S \rightarrow aSa/bSb/\epsilon$

$L = abbbba$

$S \rightarrow aSa$

$\rightarrow abSba \quad (S \rightarrow bSb)$

$\rightarrow abbSbba \quad (S \rightarrow bSb)$

$\rightarrow abbbba \quad (S \rightarrow \epsilon)$

9 Obtain a grammer to generate the following language.

$L = \{ 0^m 1^m 2^n \mid m \geq 1 \text{ and } n \geq 0 \}$

$L = \{ 01, 0011 2, \ldots \}$

~~$S \to 0S0 / 1S1 / 2S / \epsilon$~~ $S \to AB$

~~0011 2~~ $A \to 0A1 / 01$

~~$S \to 0S0$~~ $B \to 2B / \epsilon$

~~00S~~

~~001S~~ $S \to AB$

~~0011S~~ $\to 0A1B$ (~~B~~$A \to 0A1$)

~~00112~~ $\to 0011B$ ($A \to 01$)

$\to 00112B$ ($B \to 2B$)

$\to 00112$ ($B \to \epsilon$)

10. Obtain CFG for the language

$L = \{ w \mid n_a(w) = n_b(w) \}$ $\Sigma = \{a, b\}$

$L = \{$~~aa, bb~~, ab, ba, aabb, $\ldots \}$

$S \to as / bs / \epsilon$

$S \to as$

abS

$abbS$

$abbaS$

$abba\epsilon$

$S \to aSb / bSa / \epsilon$

aabb

$S \to aSb$

$\to aaSbb$

$\to aabb$

abba

$S \to aSb$

$\to abSab$

$\to$

aa $S \to as$

aas

aabs

aabbS

aabbϵ

$S \to SS$

$S \to aSbS$

$\to abSabS$

$\to abab$

$\to abSa$

$\to abba$

aababb

$S \to aSb$

$\to aaSbb$

$\to aabSabb$

$\to aababb$

Date:

** 11. Obtain CFG to generate string of balanced parenthesis.

T = {(,), {, }, [,]}

L = {(), ({}), (({})), [{}], [()], ... }

S → (S) / {S} / [S] / ∈ / SS

S → (S)
→ ((S))
→ (({S}))
→ (({[S]}))

S → SS
→ (S)S
→ ((S))S
→ (({S}))S

S → SS
S(S)
S(

S → (S)
→ (SS)
→ (S)S)
→ ((S)S)
→ (()S)
→ (){S}
→ (){[S]}
→ (){[]}

Prove W = (()())

S → (SS) / ∈
→ ((S)S)
→ (()(S))
→ (()(∈))
→ (()())

0S1 / ① / ②

000111

S → 0S1
→ 00S11
→ 000111

12 Obtain grammar to generate the language

L = {0^i 1^j | i ≠ j, i ≥ 0, j ≥ 0}

L = {011, 001, 00111, ... }

S → 0S1 / 1 / 0

S → 0S1
→ 011

S → 0S1
→ 00S11
→ 00111

$S \to 0S1$
$A \to A/B$
$A \to 0A/0$
$B \to 1B/1$

00011
$S \to 0S1$
00S11
00A11
00011

6/12/21

13. Obtain a grammar to generate a language $L = \{a^n b^{n-3} \mid n \geq 3\}$

$L = \{a^3, a^4 b, a^5 b^2, \ldots\}$

~~$S \to aaaS/aSb/$~~
$S \to aaaA$
$A \to aAb/\epsilon$
$a^5 b^2$
~~$S \to aaaA$~~
~~aaaaSb~~
~~aaaaaSbb~~
~~aaaaabb~~

$a^4 b$
$S \to aaaS$
aaaaSb
aaaab
$a^5 b^2$

$S \to aaaS$
aaaaSb
aaaaaSbb
$a^5 b^2 \epsilon$

14. Obtain a grammar to generate a language $L = \{a^n b^m / n \geq 0, m > n\}$

$L = \{b, ab^2, a^2 b^3, a^3 b^4, \ldots\}$

$S \to aSb/bS/b$
$S \to aSb$
aaSbb
$a^2 b^3$
$S \to AB$
$A \to aAb/\epsilon$
$B \to bB/b$

$S \to aSb/bS/b$
$\to aSb$
abSb
abb
$\to aSb$
aaSbb
aaaSbbb
aaabbbb
aSb
ab
aSb
aaSbb
aabbb

S → aSb/ϵ

aab
S → aSb
aaSbb
aaaSbb
aaabb

15 Obtain a grammer to generate no of a's greater than no of b's

$L = \{ w \mid n_a(w) > n_b(w) \}$

$L = \{a^2b, ba, b^2a, aba, a^3b, a^2ba, aba^2, baa\}$

$\sim L = \{ab, ab^2, \ldots\}$

S → aA
S → aab abbaa
S →

S → aA
A → aAb/ϵ/bAa

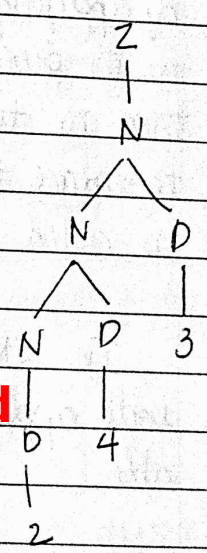
16 Obtain a grammer to generate integers.

$L = \{123, -123, +2, -1, 0, \ldots\}$

Show the derivation of unsigned integer

D → 0/1/2/3/4/5/6/7/8/9

Z → N/SN
S → +/-/ϵ
N → ND/D



17 What is the language generate by a grammer given below

S → 0A/ϵ
A → 1S

$W = \{0, 1\}$

$L = \{01, 0101, 010101, \ldots\}$

$L = \{(01)^n \mid n \geq 0\}$

S → 0A
01S
010A
0101S
0101OA

0101015
010101

⇒ $(0 \cdot 1)^*$

Date: YOUVA

18. Write the language generated by the grammer
$S \rightarrow asa \mid bsb \mid \epsilon$

$L = \{ ww^R \mid w \in (a,b)^* \}$

Derivation Tree (Parse Tree):

Let $G = (V, T, P, S)$ be a CFG, then we have two types of derivation:
→ Left most derivation (LMD)
→ Right most derivation (RMD)

Derivation is a process of driving a string or w by applying rules or productions start with start variable & in the process of derivation if we apply on RHS side if there are two or more non terminal variables than we try to replace the leftmost variable by its production, then we called derivation as left most derivation

In RMD, in the process of derivation at each step we will apply rules or productions on the right most variable.

3/12/21

1. Consider the grammer $E \rightarrow +EE \mid *EE \mid -EE \mid x \mid y$
Find LMD and RMD $w = +*-xyxy$ and write parse tree

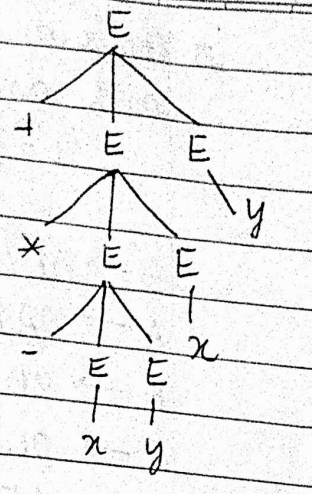
$T = \{+, -, *, x, y\}$

$V = \{E\}$

$E \rightarrow +EE$
$+*EEE$
$+*-EEEE$

+EE
+Ey
+*EEy
+*Exy
+*-EExy
+*-xyxy

→ +*-xyxy



RMD:

E → +EE

~~+E*EE~~

+EE

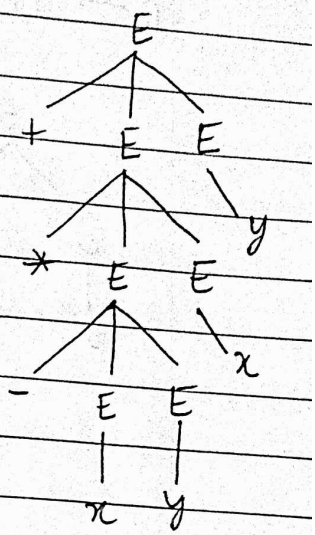
+Ey

+*EEy

+*Exy

+*-EExy

+*-xyxy



2. Construct CFG ~~for the language for the languages~~ given below

a. $L = \{a^{2n} b^m / n \geq 0, m \geq 0\}$ $w = aaaab$

Generate LMD, RMD

$L = \{aab, aabb, aaaab, aaaabb, \ldots\}$

$S \rightarrow AB$

$A \rightarrow aaA / \epsilon$

$B \rightarrow bB / \epsilon$

LMD

$S \rightarrow AB$

$aaAB$

$aaaaAB$

$aaaabB$

$aaaab$

RMD

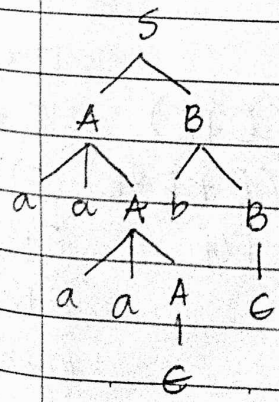
$S \rightarrow AB$

AbB

Ab

$aaAb$

$aaaab$



3 Construct CFG for the language $L = \{0^i 1^j 2^k \mid i = j \text{ or } j = k\}$ and generate LMD for the string 01122

$L = \{0122, 01122, 00112, 0111222, \ldots\}$

$S \rightarrow AB / CD$

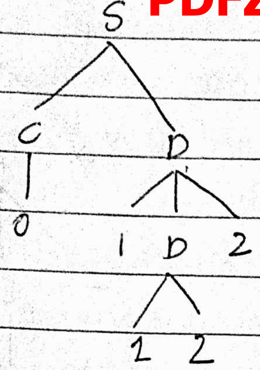
$A \rightarrow 0A1 / \epsilon 1$

$B \rightarrow 2B / \epsilon 2$

$C \rightarrow 0C / \epsilon$

$D \rightarrow 1D2 / \epsilon 12$

S → AB/CD	
0A1B	CD
011B	01D2
0112B	011D22
01122	01122



4 Consider the grammer $E \rightarrow E + E / E - E$

$E \rightarrow E * E / E / E$

$E \rightarrow (E)$

$E \rightarrow a / b / c$

LMD for (a+b*c)

RMD for (a+b)*c

$T = \{+, -, *, /, (,), a, b, c\}$

$V = \{E\}$

$E \rightarrow E * E$

LMD

$E \rightarrow (E)$

(E+E)

(a+E)

(a+E*E)

(a+b*c)

RMD

$E \rightarrow (E)$

(E+E) (E+E)

(a+E) (E+E*E)

(a+ E+

E*E

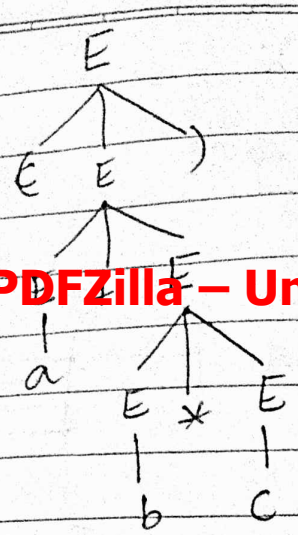
E*C

(E)*C

(E+E)*C

(E+b)*C

(a+b)*C



Ambiguous Grammar

Let $G = (V, T, P, S)$ be a CFG a grammar G is ambiguous if there exist one string $w \in T^*$ for which two or more different parse trees exists by applying either the LMD or the RMD

Ex:

$a + b * c$

$E \rightarrow E+E$

$a + E * E$

$a + b * E$

$a + b * c$

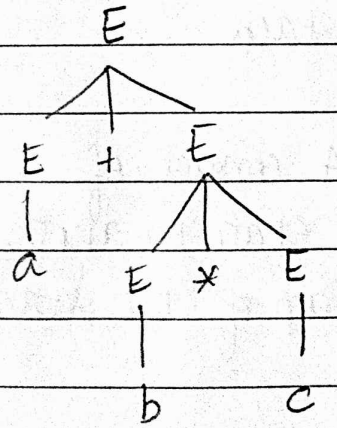
$E \rightarrow E * E$

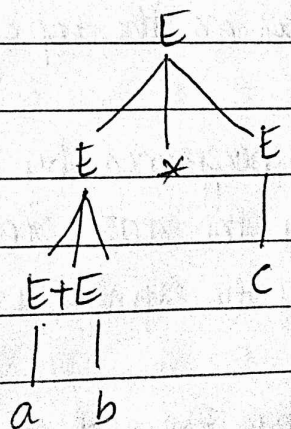
$E + E * E$

$a + E * E$

$a + b * E$

$a + b * c$





$a = 2$

$b = 3$

$c = 4$

14

20

Techniques to reduces Ambiguity:

No general purpose algorithm exist to check for ambiguity in a grammer or to remove when it is found. We often reduce ambiguity by eliminating

1. ϵ rules like $S \to \epsilon$
2. Rules with symmetric right-hand sides $S \to SS$ $E \to E+E$
3. Rules that lead to ambiguous attachment of optional postfix

1. Show that the following grammer is ambiguous & find unambiguous grammer $E \to E+E/E-E/E*E/E/E/(E)/id$ $\omega = id+id*id$

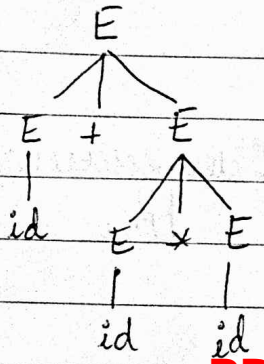
$T = \{+, -, *, /, (,), id\}$

$E \to E+E$

$id+E*E$

$id+id*E$

$id+id*id$

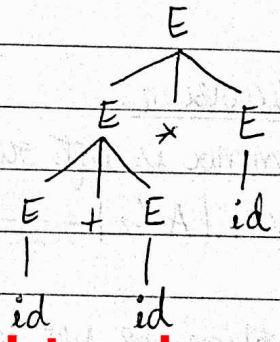


$E \to E*E$

$id*E+E*E$

$id*id+E*E$

$id+id*id$



$E \to E+T/E-T/T$

$T \to T*F/T/F/F$

$F \to (E)/id$

$id+id*id$

~~$E \to E+T$~~

~~$E \to id+T$~~

~~$E \to id+T*F$~~

~~$E \to id+F*F$~~

~~$E \to id+id*id$~~

$E \to E+T$

$\to T+T$

$\to F+T$

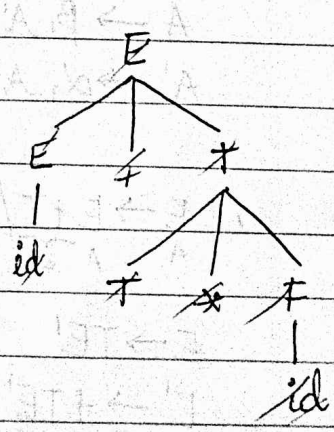
$\to id+T$

$\to id+T*F$

$\to id+F*F$

$\to id+id*F$

$\to id+id*id$



2. ST the following grammar is ambiguous & find the unambiguous grammar $S \to iCtS / iCtSeS / a$ $C \to b$ $w = ibtibtaea$

$S \to iCtS$

$ibtS$

$ibtiCtSeS$

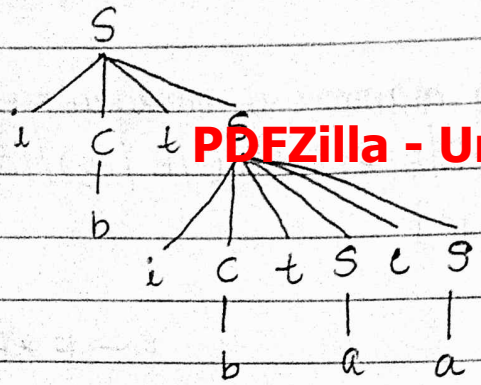
$ibtibtaea$

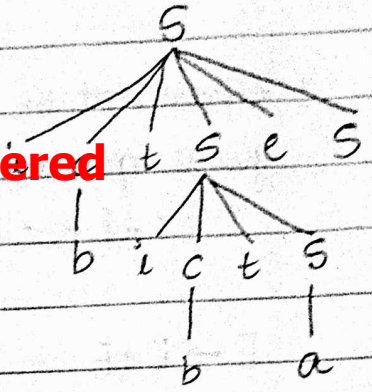
$S \to iCtSeS$

$ibtSeS$

$ibtiCtSeS$

$ibtibtaea$





$S \to iCtSM / M$

$M \to iCtMeM / a$

$C \to b$

Left Recursion:

A grammar is left recursive if it is of the following form

$A \to A\alpha_1 | A\alpha_2 | \dots | A\alpha_n | \beta_1 | \beta_2 | \beta_3 \dots | \beta_m$

Elimination of left recursion:

$A \to \beta_1 A' | \beta_2 A' | \beta_3 A' | \dots \beta_m A'$

$A' \to \alpha_1 A' | \alpha_2 A' | \dots \alpha_n A' | \epsilon$

ex: $E \to E+T / T$ ($A \to A\alpha_1 / \beta_1$)

$E \to TE'$

$E' \to +TE' / \epsilon$

$T \to T*F / F$

$T \to TF'$

$F' \to *FT' / \epsilon$

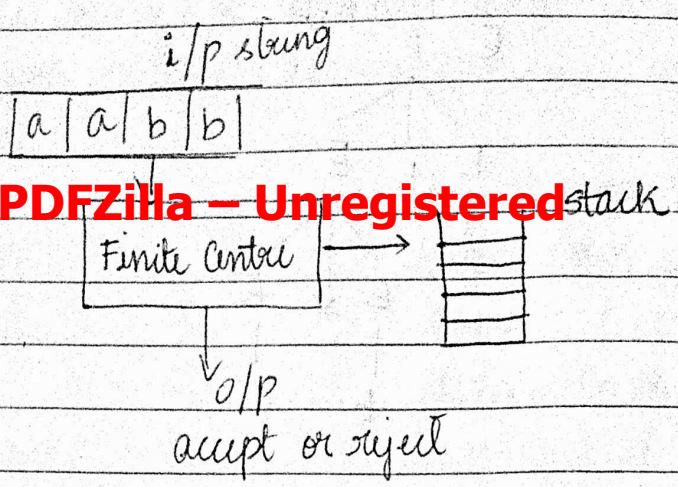
Eliminate left recursion for the following grammar

$E \to E+T/T$ — $E \to TE'$ $\quad E' \to +TE'/\epsilon$

$T \to T*F/F$ — $T \to FT'$ $\quad T' \to *FT'/\epsilon$

$T \to (E)/id$ — $\quad F \to (E)/id$

Pushdown Automata (PDA):



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

PDA is defined as $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$, where

Q - set of finite states

Σ - input alphabet which contains set of finite i/p symbols

Γ - stack alphabet which contains stack symbols.

δ - transition function defined by $\delta : Q \times \{\Sigma \cup \in\} \times \Gamma \to Q \times \Gamma^*$

q_0 - Start state

F - set of final states

The transitions performed by PDA depends on

1. Current state
2. Input symbol
3. The symbol on the top of the stack

The action performed by a PDA consist of

1. changing the state from one state to another
2. Replacing the symbol on the top of the stack.

Instantaneous Description (ID):

eg:

$(q_0, a, z_0) \vdash (q_1, z_0, A)$

$(q_1, b, A) \vdash (q_2, \in)$

PDA accepts context free languages which are defined by context free grammers

1 Construct PDA to accept the language $L = \{a^n b^n / n \geq 1\}$

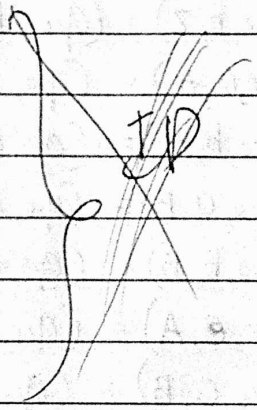
$M = (Q, \Sigma, \Gamma, \delta, q_0, F)$

$\Sigma = \{a, b\}$

$\Gamma = \{Z, A\}$

		a	b	∈
q_0	Z	Push A, q_0 ①	X	X
	A	Push A, q_0 ②	Pop A, q_1 ③	X
q_1	Z	X	X	Pop Z, q_2 ⑤
	A	X	Pop A, q_1 ④	

1) $\delta(q_0, a, Z) = (q_0, AZ)$
2) $\delta(q_0, a, A) = (q_0, AA)$
3) $\delta(q_0, b, A) = (q_1, \epsilon)$
4) $\delta(q_1, b, A) = (q_1, \epsilon)$
5) $\delta(q_1, \epsilon, Z) = (q_2, \epsilon)$



ID's:

$(q_0, aabb, Z) \vdash (q_0, abb, AZ)$
$\vdash (q_0, bb, AAZ)$
$\vdash (q_1, b, AZ)$
$\vdash (q_1, \epsilon, Z)$
$\vdash (q_2, \epsilon, \epsilon)$

$w = aabb$

not accepted

$w = a^4b^3$

$(q_0, aaaabbbb, Z) \vdash (q_0, aaabbbb, AZ)$
$\vdash (q_0, aabbb, AAZ)$
$\vdash (q_0, abbb, AAAZ)$
$\vdash (q_0, bbb, AAAAZ)$
$\vdash (q_1, bb, AAAZ)$
$\vdash (q_1, b, AAZ)$
$\vdash (q_1, \epsilon, AZ)$

Date: YOUVA

11/10/21

2. Design PDA to accept the language $L = \{wcw^R / w \in \{a,b\}^*\}$

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

$L = \{abbcbba, bacab, \ldots\}$

		a	b	c	ϵ
q_0	Z	push A, q_0	push B, q_0	X	X
	A	push A, q_0	push B, q_0	q_1, A	X
	B	push A, q_0	push B, q_0	q_1, B	X
q_1	Z	X	X	X	pop Z, q_2
	A	pop A, q_1	X	X	X
	B	X	pop B, q_1	X	X

1) $\delta(q_0, a, Z) = (q_0, AZ)$
2) $\delta(q_0, b, Z) = (q_0, BZ)$
3) $\delta(q_0, a, A) = (q_0, AA)$
4) $\delta(q_0, b, A) = (q_0, BA)$
5) $\delta(q_0, a, B) = (q_0, AB)$
6) $\delta(q_0, b, B) = (q_0, BB)$
7) $\delta(q_0, c, A) = (q_1, A)$
8) $\delta(q_0, c, B) = (q_1, B)$
9) $\delta(q_1, a, A) = (q_1, \epsilon)$

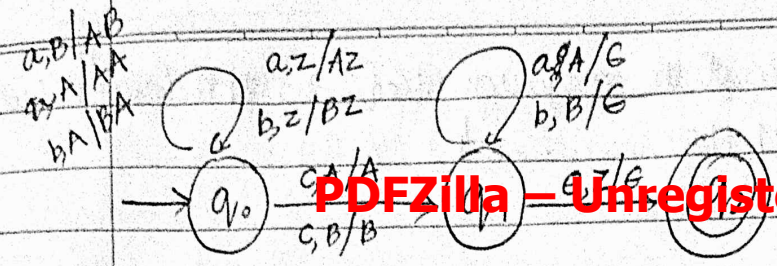
10) $\delta(q_1, b, B) = (q_1, \epsilon)$
11) $\delta(q_1, \epsilon, Z) = (q_2, \epsilon)$

IDs: $w = abbcbacab$

$(q_0, bacab) \vdash (q_0, acab, BZ)$
$\vdash (q_0, cab, ABZ)$
$\vdash (q_0, ab, AABZ)$
$\vdash (q_1, b, BZ)$
$\vdash (q_1, \epsilon, Z)$
$\vdash (q_2, \epsilon)$

$w = abcb$

$(q_0, abcb) \vdash (q_0, bcb, AZ)$
$\vdash (q_0, cb, ABZ)$
$\vdash (q_1, b, BZ)$
Not accepted



3. Design NPDA for the language $L = \{ww^R / w \in \{a,b\}^*\}$

		a	b	ϵ
	Z	push A, q_0	push B, q_0	
q_0	A	push A, q_0 / pop A, q_1	push B, q_0	
	B	push A, q_0	push B, q_0	
			pop B, q_1	
q_1	Z			pop Z, q_2
	A	pop A, q_1		
	B		pop B, q_1	

1) ~~$\delta(q_0, a, Z)$~~

Same as ② problem expect 7, 8 and add change

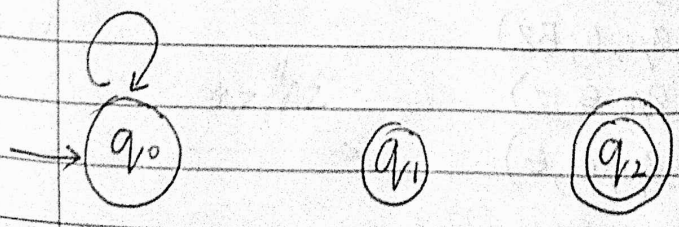
3) $\delta(q_0, a, A) = \{(q_0, AA), (q_1, \epsilon)\}$

6) $\delta(q_0, b, B) = \{(q_0, BB), (q_1, \epsilon)\}$

ID's

$(q_0, abba, z) \vdash (q_0, bba, Az)$
$\vdash (q_0, ba, BAz)$
$\vdash (q_0, a, AZ)$
$\vdash (q_1, \epsilon, Z)$
$\vdash (q_2, \epsilon, \epsilon)$

$(q_0, aba, z) \vdash (q_0, ba, Az)$
$\vdash (q_0, a, BAz)$
not accepted



13/12/21

4. Design PDA to accept the language $L = \{w \mid w \in \{0,1\}^* \# n_0(w) = n_1(w)\}$

$L = \{01, 10, 0011, 1100, \ldots\}$

		0	1	∈
	Z	push A, q_0	push B, q_0	pop Z, q_1
q_0	A	push A, q_0	pop A, q_0	
	B	pop B, q_0	push B, q_0	
q_1	Z			
	A			
	B			

1. $\delta(q_0, 0, Z) = \{(q_0, AZ)\}$
2. $\delta(q_0, 1, Z) = \{(q_0, BZ)\}$
3. $\delta(q_0, 0, A) = \{(q_0, AA)\}$
4. $\delta(q_0, 1, A) = \{(q_0, \epsilon)\}$
5. $\delta(q_0, 0, B) = \{(q_0, \epsilon)\}$
6. $\delta(q_0, 1, B) = \{(q_0, BB)\}$
7. $\delta(q_0, \epsilon, Z) = \{(q_1, \epsilon)\}$

ID's :

$w = 010110$

$(q_0, 010110, Z) \vdash (q_0, 10110, AZ)$
$\vdash (q_0, 0110, Z)$
$\vdash (q_0, 110, AZ)$
$\vdash (q_0, 10, Z)$
$\vdash (q_0, 0, BZ)$
$\vdash (q_0, \epsilon, Z)$
$\vdash (q_1, \epsilon, \epsilon)$

Date:

$w = 11010$

$(q_0, 11010, Z) \vdash (q_0, 1010, BZ)$
$\vdash (q_0, 010, BBZ)$
$\vdash (q_0, 10, BZ)$
$\vdash (q_0, 0, BBZ)$
$\vdash (q_0, \epsilon, BZ)$
not accepted

5. Design PDA to accept balanced paranthesis
Bal = $\{w \in \{), (\}^* ;$ the paranthesis are balanced $\}$
$L = \{((()))), ()(()), (\}$

		(	)	ϵ
q_0	Z	push A, q_0	X	(q_1, ϵ)
	A	push A, q_0	pop A, q_0	
	B	~~pop~~ X	X	

1. $\delta(q_0, (, Z) = \{q_0, AZ\}$
2. $\delta(q_0, (, A) = \{q_0, AA\}$
3. $\delta(q_0,), A) = \{q_0, \epsilon\}$
4. $\delta(q_0, \epsilon, Z) = \{q_1, \epsilon\}$

ID's

$w = (())$
$(q_0, (()), Z) \vdash (q_0, ()), AZ)$
$\vdash (q_0,)), AAZ)$
$\vdash (q_0,), AZ)$
$\vdash (q_0, \epsilon, Z)$
$\vdash (q_1, \epsilon)$

$w = (()$
$(q_0, (((, Z) \vdash (q_0, ((, AZ)$
$\vdash (q_0, (, AAZ)$
$\vdash (q_0, \epsilon, AZ)$
not accepted

6. Design PDA to accept string of well found paranthesis & input symbol $\Sigma = \{(,), [,], \{, \}\}$

$L = \{[(\{\})], [[]]\{()\}, \ldots\}$

		(	)	{	}	[
	z	push (, q_0		push {, q_0		push [, q_0
q_0	(	push (, q_0	pop (, q_0	push {, q_0		push [, q_0
	[	push [, q_0		push {, q_0		push [, q_0
	{	push {, q_0		push {, q_0	pop {, q_0	push [, q_0

]	∈
	pop z, q_1
pop [, q_0	

~~$w = [(\{\})]$~~

~~$(q_0, [(\{\})], z) \vdash (q_0, (\{\})$~~

1. $\delta(q_0, (, Z) = (q_0, AZ)$
2. $\delta(q_0, (, A) = (q_0, AA)$
3. $\delta(q_0, (, B) = (q_0, AB)$
4. $\delta(q_0, (, C) = (q_0, AC)$
5. $\delta(q_0,), A) = (q_0, \epsilon)$
6. $\delta(q_0, \{, Z) = (q_0, BZ)$
7. $\delta(q_0, \{, A) = (q_0, BA)$
8. $\delta(q_0, \{, B) = (q_0, BB)$
9. $\delta(q_0, \{, C) = (q_0, BC)$
10. $\delta(q_0, \}, C) = (q_0, \epsilon)$
11. $\delta(q_0, [, Z) = (q_0, CZ)$
12. $\delta(q_0, [, A) = (q_0, CA)$
13. $\delta(q_0, [, B) = (q_0, CB)$
14. $\delta(q_0, [, C) = (q_0, CC)$
15. $\delta(q_0,], C) = (q_0, \epsilon)$
16. $\delta(q_0, \epsilon, Z) = (q_1, \epsilon)$

ID's:

$(q_0, (){[]}, Z) \vdash (q_0,){[]}, AZ)$

$\vdash (q_0,){[]}, AZ)$

$\vdash (q_0, {[]}, Z)$

$\vdash (q_0, []}, BZ)$

$\vdash (q_0,]}, CBZ)$

$\vdash (q_0, }, BZ)$

$\vdash (q_0, \epsilon, Z)$

$\vdash (q_1, \epsilon, \epsilon)$

accepted