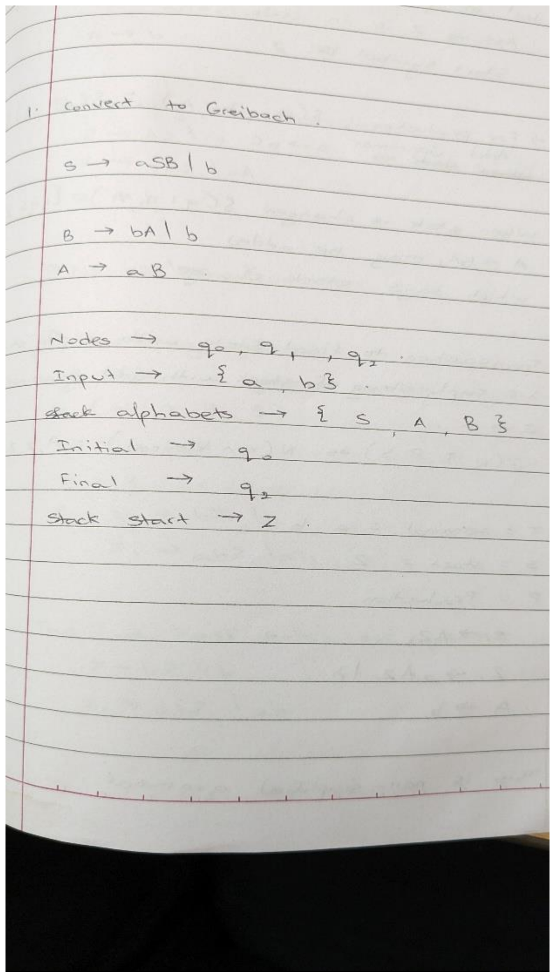
1. Following the construction of Theorem 7.1, convert this grammar to an equivalent npda:

```
S -> aSbA|b
A -> abA|λ
```



1. Convert to Greibach.

$S \rightarrow aSB \mid b$

$B \rightarrow bA \mid b$

$A \rightarrow aB$

Nodes $\rightarrow$ q_0, q_1, q_2.

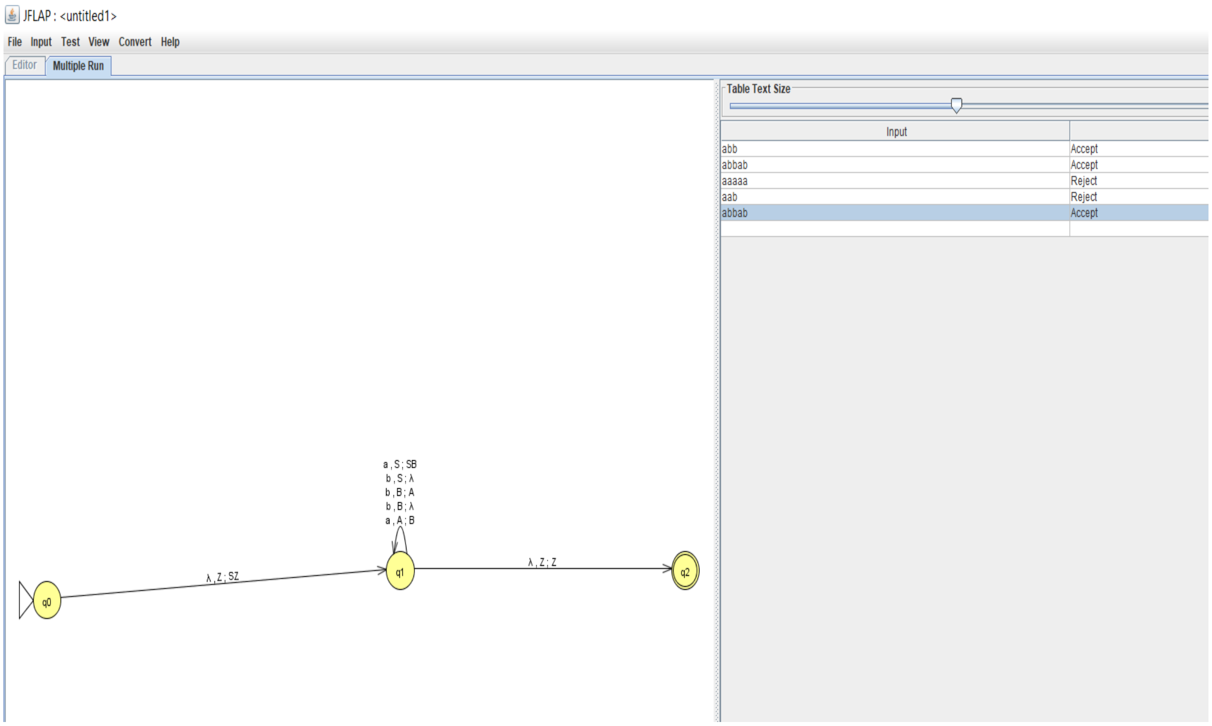
Input $\rightarrow$ $\{a, b\}$.

Stack alphabets $\rightarrow$ $\{S, A, B\}$

Initial $\rightarrow$ q_0

Final $\rightarrow$ q_2

Stack start $\rightarrow$ Z.


JFLAP : <untitled1>
File Input Test View Convert Help
Editor
Multiple Run
Table Text Size
Input
abb Accept
abbab Accept
aaaaa Reject
aab Reject
abbab Accept
a , S ; SB
b , S ; λ
b , B ; A
b , B ; λ
a , A ; B
λ , Z ; SZ
λ , Z ; Z
q0
q1
q2

2. Following the construction of Theorem 7.2, convert the following npda to an equivalent grammar. Show the initial results prior to simplification, and then the final results after simplification. You don't have to show each individual step of simplification.

Page No.
Date / /

2. Converting NPDA to CFG, following rules are followed.

→ Transaction involving pushing of start symbol of grammar on stack with help of string λ.
Assume z is in stack.
Start symbol ~~be~~ z.

→ For production, $\delta(q_1, b, A) = \{(q1, c)\}$
Add Grammar ~~when state is~~ $A \to bC$.

When state is changed $\delta(q1, b, A) = \{(q2, A)\}$
$A \to bA$, may be added.
which says stack change.

Transaction to final state with empty string
λ = empty string of stack with add grammar
$z \to \lambda$.

$G(N, T, P, S)$ as N (non terminal) $= A, z, z_1$.

T = terminal $= a, b$
S = start $= z$.
P = Production

$z \to Az_1$
$z_1 \to aAz_1 \mid \lambda$
$A \to b$.

This is non-simplified grammar.

Page No.
Date

Now, simplification includes -

1. Removing Null production by inserting new production which is done by inserting empty string to all other grammar.

$Z \rightarrow AZ_1 \mid A$

$Z_1 \rightarrow aAZ_1 \mid aA$

$A \rightarrow b$

2. Removing Unit Production

$Z \rightarrow AZ_1 \mid b$

$Z_1 \rightarrow aAZ_1 \mid aA$

$A \rightarrow b$

3. No useless Production

4. Shortening grammar

$A \rightarrow b$ $\therefore$ grammar can be reduced.

$G'(N, T, P, S)$

$N = a, b$

$S = Z$

$P = Z \rightarrow bZ_1 \mid b$

$Z_1 \rightarrow abZ_1 \mid ab$

Thus, we have solved our question

$Z \rightarrow bZ_1 \mid b$

$Z_1 \rightarrow abZ_1 \mid ab$

3. Prove that the language L with Σ = {a}, where L = {a^n : n is a power of 2} is not context-free.

Page No.
Date / /

3. L with $\Sigma = \{a\}$, where $L = \{a^n : n$ is power of 2$\}$ is not a CF.
L produces language L such that
$L = \{a^{n*2} : n$ is a power of 2$\}$.

i.e. L = {a, aaaa, aaaaaaaa, ... }

Using Pumping Lemma to prove given language is ≠ CFL.
$|x| \geq n$.
where x = length of input alphabet
→ uvwxyz ∈ L and i >= 0.
; where z = uvwxy

$|uvx| <= |x|$
$|vx| != 0$.

If there exists atleast 1 value for i such that
$uv^i wx^i y$ does ~~n~~xt ∈ L,
then L ≠ CFL
L with $\Sigma = \{a\}$.

L = {a, aaaa, aaaaaaaa, ... }
Let z = aaaaaaaa.
and u = a, v = aa, w = aa,
x = aa, y = a.

Page No.

Date

then $a(aa)^i aa(aa)^i, a \in L$ and $i > 0$.
for $i = 0$, it will give aaaa which does not $\in L$ ... $\therefore L \neq CFL$.

OR

Let $z = a^k$, k is power of 2.
$u = a^{k-4}$, $v = a^i$, $w = a^1$,
$x = a^1$, $y = a^1$;
then $uv^i w x^i y \in L$ and $i >= 0$.

for $i = 0$
a^{k-4} aa which is a^{k-2} and a^k is not sum.
$\therefore k^{-2}$ does not $\in L$.
$\therefore L \neq CFL$.

4. Let L be the language, there exists an
with $|x| >= n$,
for all $x \in L$ ... b $c\}^*$.

4. Prove that the language L = {$a^x b^y c^z$: z > y, z > x} is not context-free.

Let $z = a^n$, k is power of 2.

$u = a^{k-4}$, $v = a^1$, $w = a^1$

$x = a^1$, $y = a^1$;

then $uv^i wx^i y \in L$ and $i >= 0$.

for $i = 0$

$a^{k-4}\, a\, a$ which is a^{k-2} and a^k is not

sum.

∴ k^{-2} does not $\in L$.

∴ $L \neq CFL$.

4. Let L be the language, there exists an integer n i.e. for all $x \in L$ with $|x| >= n$ there exists $u, v, w, x, y \in \{a, b, c\}^*$.

let $z = uvwxy$.

- $|vwx| <= n$
- $|wx| >= 1$
- for all $i > 0$, $uv^i wx^i y \in L$.

Given $L = \{a^x b^x c^z : x > y, z > x\}$.

Page No.
Date

Let $x = 2$, $y = 2$, $z = 3$.

String $x = a^2 b^2 c^3 = aabbccc$.

$|x| = |aabbccc| = 7$.

Let $N = 5$.

$\therefore |x| \geq N$.

$x = \underbrace{a}_{u}\,\underbrace{a}_{v}\,\underbrace{bb}_{w}\,\underbrace{c}_{x}\,\underbrace{cc}_{y}$.

$\rightarrow |vwx| = |abbcc| = 5 \leq N$.

$|vx| = |abc| = 3 \geq 1$.

$i = 2$.

$uv^i w x^i y$

$= a(ab)^2 bc(c)^2 c$

$= a^2 bab^2 c^4 \notin L$.

$\therefore$ Language L is not context free.

5. Let language L = {a,b,c}* : $n_a(w) = n_b(w) = n_c(w)$}. Prove that the complement of L is context-free.

Page No.
Date

5. Given language -

$L = \{ \{a,b,c\}^* : n_a(\omega) = n_b(\omega) = n_c(\omega) \}$

$L' = \{ \{a,b,c\}^* : n_a(\omega) \neq n_b(\omega) \} \cup$
$\{a,b,c\}^* : n_b(\omega) \neq n_c(\omega) \} \cup$
$\{a,b,c\}^* : n_a(\omega) \neq n_c(\omega) \} \}$

The above language is

S → RAR | TBT | UCU

R → cR | G

T → aT | G

U → GU | E

A → D | E

D → FGE | OO

E → FMF | EE

F → aFb | bFa | FF | G

I → kHk | II

J → kHk | JJ

K → bkG | ckb | kk | ∈

L → N | P

N → QGQ | NN

P → QMQ | PP

Q → aQc | cQa | QQ | c

G → aG | a

H → bH | b

M → cM | c .

where, variable set = {A, B, C, ..., U}

~~[illegible]~~ start symbol ⇒ S .

Page No.
Date

Since there exists a CFG for the complement of language, hence it is proved that complement of language is context free.

Regular L is subset of CFL, which means regular L and FA machine which accept CFL are PDA. So if language can observe FA, then it is cyclic & if it can form PDA, then it is called CFL.

We need to check for CFL as RL

$L = \{a, b, c\}^* : n_a(w) = n_b(w) = n_c(w)\}$

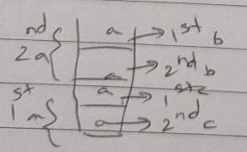
→ Also contains empty string

eg- aabbcc.

We push 'a' into stack or non ~~number~~ 'b', we can pop 'a' from start.

This way we can compare 'a' & 'b' when c comes, the stack becomes empty.

Another way, we add 'a' and 'aa' into stack.



∴ the above is not CF.

Page No.
Date

L complement

$\rightarrow \{ \{a, b, c\}^* : n_a(w) \neq n_b(w) \neq n_c(w) \}$

∴ Letters are independent & do not have same value.

eg - abbccc.

We can define a regular expression for $a^* b^* c^*$ which means its Regular languages.

RL is CFL as its subset.

Hence Proved.