Context-Free Languages

Generator — Representator Automata

Context-Free Grammar — Push-down Automata

Automaton ⟷ stack

Grammars:

Grammars express languages.

Ex:

<sentence> → <noun-phrase> <predicate>
<noun-phrase> → <article> <noun>
<predicate> → <verb>
<article> → a
<article> → the
<noun> → cat
<noun> → dog
<verb> → runs
<verb> → walks

Variables (or) Non-terminals

Terminals

Production Rules

Derivation of "The dog walks"

<sentence> → <noun-phrase> <predicate>
→ <noun-phrase> <verb>
→ <article><noun> <verb>
→ the <noun> <verb>
→ the dog <verb>
→ the dog walks.

Language of the Grammar :

L = { "a cat runs", a dog runs
a cat walks, a dog walks
the cat runs , the dog runs
the cat walks , the dog walks }

A CFG is defined as $G = (V, T, P, S)$ where

V is a set of non-terminals

T is a set of terminals

P is a set of productions of the form

$$A \rightarrow \alpha$$

where A is a non-terminal and
α is a string over $(V \cup T)^*$

S is the start symbol.

Consider the productions shown below:

$$S \rightarrow aCa$$
$$C \rightarrow aCa \,|\, b.$$

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

$V = \{S, C\}$

$T = \{a, b\}$

$P = \{ \; S \rightarrow aCa$
$\quad C \rightarrow aCa \,|\, b$
$\}$

S is the start symbol.

* Let $G = (V, T, P, S)$ be a Grammar. Then the set $L(G) = \{w \in T^* : S \overset{*}{\Rightarrow} w\}$ is the language generated by G.

* The strings which contain variables as well as terminals are called sentential forms of the derivation.
(or) sequential forms.

* Derivation Tree. (or) Parse Tree * Yield.

1. Consider the grammar $G = (\{S\}, \{a, b\}, S, P)$ and P is given by

$$S \to aSb \mid \epsilon$$

$\overset{*}{\Rightarrow}$ derivation

$\overset{*}{\underset{lm}{\Rightarrow}}$ left most derivation

$\overset{*}{\underset{rm}{\Rightarrow}}$ rightmost derivation

w_1 $S \Rightarrow aSb$ w_2

$\Rightarrow aaSbb$ w_3

$\Rightarrow aabb$ w_4

$w_1 \Rightarrow w_2 \Rightarrow w_3 \Rightarrow \cdots w_n$

$w_1 \overset{*}{\Rightarrow} w_n$.

$$\boxed{S \overset{*}{\Rightarrow} aabb}$$

$w_1 \overset{*}{\Rightarrow} w_4$.

aabb is the string in the CFL generated by CFG.

aaSbb is a sentential form.

<u>Derivations</u>:

$S \overset{*}{\Rightarrow} \epsilon$ $S \overset{*}{\Rightarrow} ab$ $S \overset{*}{\Rightarrow} aabb$ $S \overset{*}{\Rightarrow} aaabbb$

$\therefore$ Language $= \{a^n b^n \mid n \geq 0\}$.

2. Consider the Grammar $G = (N, T, P, S)$ $V = \{S\}$

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

$P = \{S \to aSa \mid bSb \mid c\}$. Write the language generated by the Grammar.

$S \overset{*}{\Rightarrow} c$ $S \overset{*}{\Rightarrow} aca$ $S \overset{*}{\Rightarrow} bcb$ $S \overset{*}{\Rightarrow} abcba$

$S \overset{*}{\Rightarrow} bacab$ $S \overset{*}{\Rightarrow} aacaa$ $S \overset{*}{\Rightarrow} bbcbb$

$S \overset{*}{\Rightarrow} abacaba$

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

Scanned by CamScanner

③. $G = (V, T, P, S)$

$V = \{S, C\}$ $T = \{a, b\}$

$P = \{$

$S \rightarrow aCa$

$C \rightarrow aCa \mid b$

$\}$

S is the start symbol. What is the language generated by this grammar?

$S \overset{*}{\Rightarrow} aba$ $S \overset{*}{\Rightarrow} aabaa$ $S \overset{*}{\Rightarrow} aaabaaa$

$L(G) = \{a^n b a^n \mid n \geq 1\}$.

④. $L = \{aa, ab, ba, bb\}$ CFG = ?.

R.E = $(a+b)(a+b)$

$P \Rightarrow$

$S \rightarrow AA$ S start symbol

$A \rightarrow a \mid b$ $T = \{a, b\}$ $N = \{S, A\}$

⑤. $L = \{a^n \mid n \geq 0\}$

$\{a, aa, aaa, \ldots\}$

$\Rightarrow$ Right Recursion. Left Recu

$P \Rightarrow$

$S \rightarrow$ ~~a~~ $aS \mid \varepsilon$ (or) $S \rightarrow Sa \mid \varepsilon$

$T = \{a\}$ $N = \{S\}$ S start symbol.

⑥. $(a+b)^*$

$S \rightarrow aS \mid bS \mid \varepsilon$

⑦. Atleast 2 length $(a+b)(a+b)(a+b)^*$

$S \rightarrow AAB$

$A \rightarrow a \mid b$

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

(8). Strings having length atmost 2.

$(a+b+\varepsilon)(a+b+\varepsilon)$.

$S \rightarrow AA$

$A \rightarrow a|b|\varepsilon$

(9). $a(a+b)^* b$

$S \rightarrow aAb$

$A \rightarrow aA|bA|\varepsilon$

(10). Strings starting and ending with diff. symbols.

Reg. Exp $a(a+b)^* b + b(a+b)^* b$.

$S \rightarrow aAb | bAa$

$A \rightarrow aA | bA | \varepsilon$

(11). Strings starting and ending with same symbols.

Reg. Exp. $a(a+b)^* a + b(a+b)^* b$.

$S \rightarrow aAa | bAb | a | b | \varepsilon$.

$A \rightarrow aA | bA | \varepsilon$.

(12). $a^n b^n \mid n \geq 1$.

$S \rightarrow aSb \mid ab$.

(13). $ww^R \cup waw^R \cup wbw^R$. $w \in (a+b)^*$.

$S \rightarrow aSa | bSb | a | b | \varepsilon$.

(14). Even-length strings $((a+b)(a+b))^*$

$S \rightarrow BS | \varepsilon$

$B \rightarrow AA$

$A \rightarrow a|b$

(15). $a^n b^m \mid n, m \geq 1$.

$S \rightarrow AB$

$A \rightarrow aA | a$

$B \rightarrow bB | b$

Scanned by CamScanner

16. $a^n b^n c^m \mid n,m \geq 1$

S → AB
A → aAb | ab
B → cB | c

17. $c^m a^n b^n \mid n,m \geq 1$

S → BA
A → aAb | ab
B → cB | c

18. $a^n c^m b^n \mid n,m \geq 1.$

A → aBb | aAb
B → cB | c

19. $a^n b^n c^m d^m \mid n,m \geq 1$

S → AB
A → aAb | ab
B → cBd | cd

20. $a^n b^{2n} \mid n \geq 1$

S → aSbb | abb

21. $a^n b^m c^m d^n \mid n,m \geq 1$

S → aSd | aAd
A → bAc | bc

22. $a^{m+n} b^m c^n \mid m,n \geq 1$

$a^n a^m b^m c^n$

S → aSc | aAc
A → aAb | ab

23. $a^n b^{n+m} c^m \mid n,m \geq 1$

$a^n b^n b^m c^m \mid n,m \geq 1$

S → AB
A → aAb | ab
B → bBc | bc

24. $a^n b^m c^{n+m} \mid n,m \geq 1$

$a^n b^m c^m c^n \mid n,m \geq 1$

S → aAc | aSc
A → bAc | bc

25. $a^n b^n c^n \mid n \geq 1$

26. $a^n b^m c^n d^m \mid n,m \geq 1$

⇓

CFG is not possible

27. $a^i b^j c^k \mid i = j + k$

~~S → aSc | ε~~

S → aSc | T
T → aTb | ε

28. $\{ a^i b^j c^k \mid j = i$ or $j = k \}$

S → AT | UC
A → aA | ε
C → cC | ε
T → bTc | ε
U → aUb | ε

(29). $\{a^i b^j c^k \mid i = j \text{ or } i = k\}$

$S \to TC \mid U$

$C \to cC \mid \epsilon$

$T \to aTb \mid \epsilon$

$U \to aUc \mid B$

$B \to bB \mid \epsilon$

(30). $\{a^i b^j c^k \mid i < j \text{ or } i > k\}$

$S \to TBC \mid AU$

$T \to aTb \mid \epsilon$

$B \to bB \mid b$ ⇒ any no. of b

$C \to cC \mid \epsilon$ ⇒ any no. of c

$A \to aA \mid a$ ⇒ any no. of i

$U \to aUc \mid V$

$V \to bV \mid \epsilon$

(31). $\{a^i b^j \mid i \le j \le 2i\}$

$S \to aSb \mid aSbb \mid \epsilon$

Why the name context-free Grammar?.

$A \to aAb \mid ab$

$A \Rightarrow aAb$
$\Downarrow$
$\Rightarrow a\,ab\,b$ (irrespective of the preceeding & following symbols A can be replaced with aAb or ab).

so when the productions

$A \to \alpha$ $\quad A \in V$
$\quad \alpha \in (V \cup T)^*$

is called context-free Grammar.

Incase..

if i have a productions $aA \to aa$

then if A has a preceeding then

only it will be replaced by aa.

CFG:

Can be :

Ambiguous & Unambiguous.
Deterministic & Non-deterministic.
Left-Recursion & Right-Recursion

1. Consider the grammar

$$S \rightarrow aS \mid aSbS \mid \epsilon$$

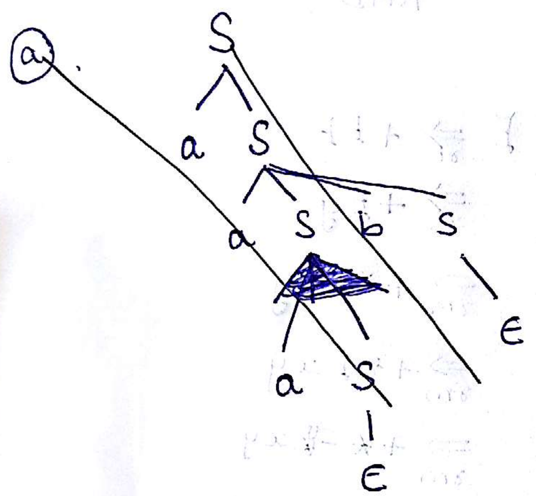
The grammar is ambiguous. Show in particular that the string aab has two:

a). Parse trees.

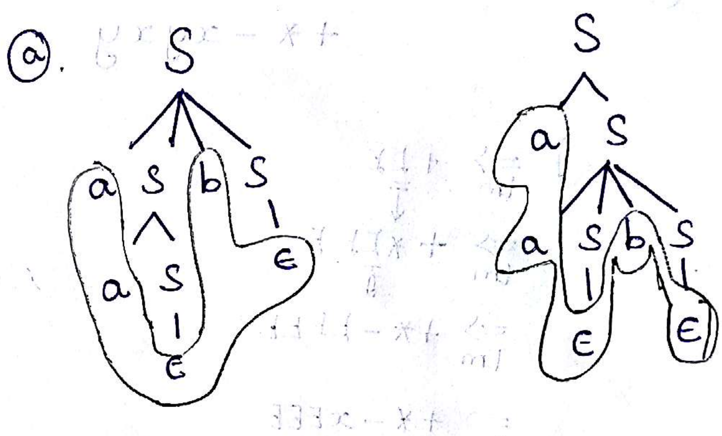
b). LMD

c). RMD

a).



a).



b).

$$S \underset{lm}{\Rightarrow} aS \underset{lm}{\Rightarrow} aaSbS \underset{lm}{\Rightarrow} aabS \underset{lm}{\Rightarrow} aab$$

$$S \underset{lm}{\Rightarrow} aSbS \underset{lm}{\Rightarrow} aaSbS \underset{lm}{\Rightarrow} aabS \underset{lm}{\Rightarrow} aab$$

c).

$$S \underset{rm}{\Rightarrow} aS \underset{rm}{\Rightarrow} aaSbS \underset{rm}{\Rightarrow} aaSb \underset{rm}{\Rightarrow} aab$$

$$S \underset{rm}{\Rightarrow} aSbS \underset{rm}{\Rightarrow} aSb \underset{rm}{\Rightarrow} aaSb \underset{rm}{\Rightarrow} aab$$

② The following grammar generates prefix expressions with operands x and y and binary operators $+$, $-$ and $*$:

$$E \rightarrow +EE \mid *EE \mid -EE \mid x \mid y$$

ⓐ Find LMD & RMD and derivation tree for the string $+*-xyxy$

ⓑ Prove that this grammar is unambiguous.

ⓐ LMD $\qquad +*-xyxy \qquad$ RMD

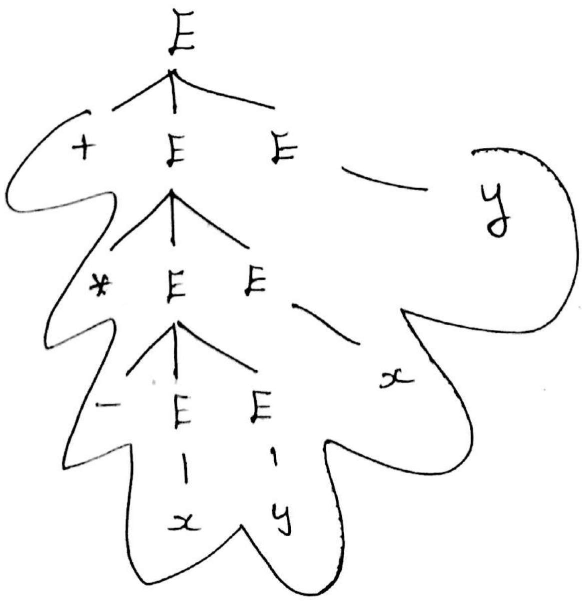
LMD

$$E \underset{lm}{\Rightarrow} +EE$$
$$\underset{lm}{\Rightarrow} +*EEE$$
$$\underset{lm}{\Rightarrow} +*-EEEE$$
$$\underset{lm}{\Rightarrow} +*-xEEE$$
$$\underset{lm}{\Rightarrow} +*-xyEE$$
$$\underset{lm}{\Rightarrow} +*-xyxE$$
$$\underset{lm}{\Rightarrow} +*-xyxy$$

RMD

$$E \underset{rm}{\Rightarrow} +EE$$
$$\underset{rm}{\Rightarrow} +Ey$$
$$\underset{rm}{\Rightarrow} +*EEy$$
$$\underset{rm}{\Rightarrow} +*Exy$$
$$\underset{rm}{\Rightarrow} +*-EExy$$
$$\underset{rm}{\Rightarrow} +*-Eyxy$$
$$\underset{rm}{\Rightarrow} +*-xyxy.$$

Derivation Tree



Simplified Forms and Normal Forms:

(*1). Elimination of Nullable Variables

(*2). Elimination of Unit productions

(*3). Elimination of Useless variables

Elimination of Nullable Variables:

Nullable Variables:

A nullable variable in a CFG $G = (V, \Sigma, S, P)$ is defined as follows...

(*) Any variable A for which P contains the production $A \to \varepsilon$ is nullable.

(*) If P contains the production $A \to B_1 B_2 \ldots B_n$ and $B_1, B_2, \ldots B_n$ are nullable variables, then A is nullable.

(*) No other variables in V are nullable.

Note:

If $\varepsilon \in L(G)$ then $S \to \varepsilon$ S cannot be removed.

Problems:

(1). $S \to aSb \mid aAb$

$A \to \varepsilon$

Nullable variable: A

$S \to aSb \mid aAb \mid ab$.

(2) Step.

(*1) Identify nullable variable

(*2) In the production rule have the productions with & without nullable variable (R.H.S) & remove the rule $A \to \varepsilon$

Note: While eliminating the context or meaning of the Grammar should not change.

(2). $S \to AB$

$A \to aAA \mid \varepsilon$

$B \to bBB \mid \varepsilon$.

Nullable variable : A & B.

$S \to AB \mid A \mid B \mid \varepsilon$

$A \to aAA \mid aA \mid a$

$B \to bBB \mid bB \mid b$

(3). $S \to AbaC$

$A \to BC$

$B \to b \mid \varepsilon$

$C \to D \mid \varepsilon$

$D \to d$

Nullable variable : A, B & C

$S \to AbaC \mid Aba \mid baC \mid ba$

$A \to BC \mid B \mid C$

$B \to b$

$C \to D$

$D \to d$

(4). $S \to aA$

$A \to b \mid \varepsilon$

Nullable variable : A

$S \to aA \mid a$

$A \to b$

(5). $S \to ABAC$

$A \to aA \mid \varepsilon$

$B \to bB \mid \varepsilon$

$C \to c$

Nullable variable : A & B

$S \to ABAC \mid ABC \mid BAC \mid AC \mid AAC \mid BC \mid C$

$A \to aA \mid a$

$B \to bB \mid b$

$C \to c$

(6). $S \to aSa$

$S \to bSb \mid \varepsilon$

Nullable variable : S

$S \to aSa \mid aa \mid bSb \mid bb \mid \varepsilon$

(7). $S \to a \mid Xb \mid aYa$

$X \to Y \mid \varepsilon$

$Y \to b \mid X$.

Nullable variable : X , Y

$S \to a \mid Xb \mid aYa \mid b \mid aa$

$X \to Y$

$Y \to b \mid X$

⑧. Design a CFG for regular expression $r = (a+b)^* bb (a+b)^*$, which is free from ϵ-productions.

$S \rightarrow AbbA$

$A \rightarrow aA \mid bA \mid \epsilon$

Nullable variable: A

$S \rightarrow AbbA \mid Abb \mid bbA \mid bb$

$A \rightarrow aA \mid bA \mid a \mid b$

Algorithm: (Finding an equivalent CFG with no ϵ-production):

Given a CFG $G = (V, T, P, S)$,

Construct a CFG $G_1 = (V, T, P_1, S)$ with no ϵ-productions as follows.

(*1). Initialize P_1 to be P.

(*2). Find all nullable variables in V,

(*3). For every production $A \rightarrow \alpha$ in P, add to P_1 every production that can be obtained from this one by deleting from α one or more of the occurences of nullable variables in α.

(*4). Delete all ϵ-productions from P_1. Also delete any duplicates, as well as productions of the form $A \rightarrow A$.

Elimination of Unit Productions :

Algorithm 2: (Finding an equivalent CFG with no unit productions)

Given a CFG (V, T, P, S) with no ε-productions, construct a grammar $G_1 = (V, T, P_1, S)$ having no unit productions as follows :

(1). Initialize P_1 to be P.

(2). For each $A \in V$, find the set of A-derivable variables.

(3). For every pair (A, B) such that B is A-derivable and every nonunit production $B \to \alpha$, add the production $A \to \alpha$ to P_1 if it is not already present in P_1.

(4). Delete all unit productions from P_1.

(1).

$S \to Aa \mid B$

$B \to A \mid bb$

$A \to a \mid bc \mid B$.

$B \to \cancel{A} \to a \mid bc \mid \cancel{B}$ $(B \to A)$

$A \to \cancel{B} \to \cancel{A} \mid bb$ $(A \to B)$

$S \to B \to bb \mid a \mid bc$ $(S \to B)$

$S \to Aa \mid bb \mid a \mid bc$

$B \to bb \mid a \mid bc$

$A \to a \mid bc \mid bb$

(2).

$S \to AB$

$A \to a$

$B \to c \mid b$

$C \to D$

$D \to E$

$E \to a$

$S \to AB$

$A \to a$

$B \to b \mid a$

$C \to a$

$D \to a$

$E \to a$

$B \to \cancel{C} \to \cancel{D} \to \cancel{E} \to a$ $(B \to C)$

$C \to \cancel{D} \to \cancel{E} \to a$ $(C \to D)$

$D \to E \to a$ $(D \to E)$

Derivable :

① If $A \to B$ is a production, B is A-derivable.

② If C is A-derivable, $C \to B$ is a production, and $B \neq A$, then B is A-derivable.

③ No other variables are A-derivable.

④ A variable A is A-derivable only if $A \to A$ is actually a production.

③. $S \to S+T \mid T$

$T \to T*F \mid F$

$F \to (S) \mid a$.

$S \to T$

$T*F \mid F$

$(S) \mid a$

$T \to T*F \mid F$

$F \to (S) \mid a$.

$S \to S+T \mid T*F \mid (S) \mid a$

$T \to T*F \mid (S) \mid a$

$F \to (S) \mid a$

④. $I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$F \to I \mid (E)$

$T \to F \mid T*F$

$E \to T \mid E+T$.

$I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$F \to (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$T \to T*F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$E \to E+T \mid T*F \mid (E) \mid a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

(5). $S \to A \mid bb$

$A \to B \mid b$

$B \to S \mid a$.

$S \to A \to B \mid b$

$S \mid a$

$A \to B \mid b$

$S \mid a \mid b$

$A \mid bb$

$B \to S \mid a$

$A \mid bb \mid a$

$B \mid b$

$S \to bb \mid a \mid b$

$A \to b \mid bb \mid a$

$B \to a \mid bb \mid b$

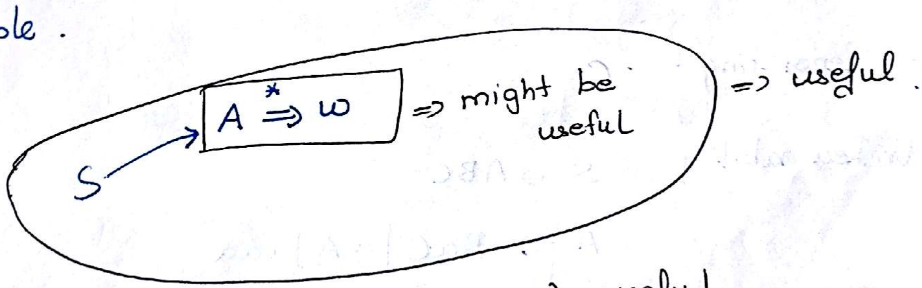
Elimination of Useless Variables:

Elimination of Useless Variables involves two things...

(*1). We say X is generating if $X \stackrel{*}{\Rightarrow} w$ for some terminal string w. Note that every terminal is generating, since w can be that terminal itself, which is derived by zero steps.

(*2). We say X is reachable if there is a derivation $S \stackrel{*}{\Rightarrow} \alpha X \beta$ for some α and β.

Useful symbol will be both generating and reachable.



$A \stackrel{*}{\Rightarrow} aBb \Rightarrow$ not useful.

(1). $S \to AB \mid a$

$A \to BC \mid b$

$B \to aB \mid C$

$C \to aC \mid B$.

Not Generating:

Useless variables: B, C

$S \to AB \mid a$

$A \to BC \mid b$

On Reachablity: $S \to AB$ not reachable

$A \to BC$ not reachab.

$A \to b$ no use

$\boxed{S \to a}$.

② . $S \rightarrow AB \mid AC$

$A \rightarrow aAb \mid bAa \mid a$

$B \rightarrow bbA \mid aaB \mid AB$

$C \rightarrow abCA \mid aDb$

$D \rightarrow bD \mid aC$.

Useless Variables :

Not. Generating : C, D

Un.Reachability : $S \rightarrow AC$

$\therefore$
$$\boxed{\begin{array}{l} S \rightarrow AB \\ A \rightarrow aAb \mid bAa \mid a \\ B \rightarrow bbA \mid aaB \mid AB \end{array}}$$

③. $S \rightarrow ABC \mid BaB$

$A \rightarrow aA \mid BaC \mid aaa$

$B \rightarrow bBb \mid a$

$C \rightarrow CA \mid AC$.

Useless Variables :

Not. Generating : C

Un Reachability $S \rightarrow ABC$

$A \rightarrow BaC \mid aA \mid aaa$

$\therefore$
$$\boxed{\begin{array}{l} S \rightarrow BaB \\ B \rightarrow bBb \mid a \end{array}}$$

④. $S \rightarrow AB \mid a$

$A \rightarrow b$

Useless variables : Not Generating : B

Un-Reachability : $S \rightarrow AB$

$A \rightarrow b$

$\therefore$ $\boxed{S \rightarrow a}$

⑤. S →

A →

B →

X →

Useless varia

∴

⑥. A →

X →

Y →

Z → Z

Useless var

⑦. S →

A →

B →

C →

Useless va

No

Un

Scanned by CamScanner

⑤. $S \to aB \mid bX$

$A \to BAd \mid bSX \mid a$

$B \to aSB \mid bBX$

$X \to SBD \mid aBx \mid ad$.

Useless variables:

Not. Generating : D, B.

Unreachable : $S \to aB$

$A \to BAd \mid bSX \mid a$

$B \to aSB \mid bBX$

$X \to SBD \mid aBx$.

$\therefore$ $\boxed{\begin{array}{l} S \to bX \\ X \to ad \end{array}}$

⑥. $A \to xyz \mid Xyzz$

$X \to Xz \mid xYz$

$Y \to yYy \mid Xz$

$Z \to Zy \mid z$

Useless variables:

Not. Generating : $\cancel{Z}$ X, Y

Unreachable : $Z \to Zy \mid Z$

$\therefore$ $\boxed{A \to xyz}$

⑦. $S \to aC \mid SB$

$A \to bSCa$

$B \to aSB \mid bBC$

$C \to aBC \mid ad$.

$\therefore$ $\boxed{\begin{array}{l} S \to aC \\ C \to ad \end{array}}$

Useless variables:

Not. Generating : B

Un.reachable : $A \to bSCa$

$\to a$

Chomsky Normal Form:

A CFG is in CNF if every production is of one of these two types:

$$A \to BC$$
$$A \to \alpha$$

where A, B and C are variables and a is a terminal symbol.

Converting a CFG to CNF:

1. Eliminating ε-productions.
2. Eliminating unit productions.
3. Eliminating Useless symbols.
4. Restricting the right sides of productions to single terminals or strings of two ~~or more~~ variables.

1. Convert the below grammar to CNF:

$S \to ASB \mid \varepsilon$
$A \to aAS \mid a$
$B \to SbS \mid A \mid bb$

1. Elimate ε-production.,.

$S \to ASB \mid AB$
$A \to aAS \mid a \mid aA$
$B \to SbS \mid A \mid bb \mid Sb \mid bS \mid b$.

2. Eliminate unit-production.,.

$S \to ASB / AB$
$A \to aAS \mid a \mid aA$
$B \to SbS \mid bb \mid sb \mid bS \mid b \mid aAS \mid a \mid aA$

3. No. Useless symbols.,.

④. Restricting the right sides of productions to single terminals or strings of 2 variables.

$S \rightarrow X_1 B \mid AB$

$A \rightarrow X_a X_1 \mid X_a \mid X_a A$

$B \rightarrow S X_2 \mid X_b X_b \mid S X_b \mid X_b S \mid X_b \mid X_a X_1 \mid X_a \mid X_a A$

$X_1 \rightarrow AS$

$X_2 \rightarrow X_b S$

$X_a \rightarrow a$

$X_b \rightarrow b$

②. $S \rightarrow AACD$

$A \rightarrow aAb \mid \varepsilon$

$C \rightarrow aC \mid a$

$D \rightarrow aDa \mid bDb \mid \varepsilon$

① Eliminate ε-production.,.

$S \rightarrow AACD \mid ACD \mid CD \mid AAC \mid AC \mid C$

$A \rightarrow aAb \mid ab$

$C \rightarrow aC \mid a$

$D \rightarrow aDa \mid bDb \mid aa \mid bb$

②. Eliminate Unit production.,.

$S \rightarrow AACD \mid ACD \mid CD \mid AAC \mid AC \mid aC \mid a$

$A \rightarrow aAb \mid ab$

$C \rightarrow aC \mid a$

$D \rightarrow aDa \mid bDb \mid aa \mid bb.$

③. No. Useless s/ms.

④. Restricting Right sides of productions to single terminals or strings of 2 variable.

$S \rightarrow X_2 X_1 \mid A X_1 \mid CD \mid X_2 C \mid AC \mid X_a C \mid a$

$A \rightarrow X_3 X_b \mid ab$

$C \rightarrow X_a C \mid a$

$D \rightarrow X_4 X_a \mid X_5 X_b \mid \cancel{X_a X_a} \mid \cancel{bb}\, X_b X_b$

$X_2 \rightarrow AA$

$X_1 \rightarrow CD$

$X_a \rightarrow a$

$X_b \rightarrow b$

$X_3 \rightarrow X_a A$

$X_4 \rightarrow X_a D$

$X_5 \rightarrow X_b D$

$\boxed{\text{OR}}$ P.T.O.

Hint: $A \to BCDBCE$

$A \to BY_1$
$Y_1 \to CY_2$
$Y_2 \to DY_3$
$Y_3 \to BY_4$
$Y_4 \to CE$

4. $S \to AT_1$ $S \to AU_1$ $S \to AV_1$
$T_1 \to AT_2$ $U_1 \to CD$ $V_1 \to AC$
$T_2 \to CD$

$S \to CD \mid AC \mid X_a C \mid a$

$A \to X_a W_1$ $W_1 \to A X_b$
$A \to X_a X_b$ $Y_1 \to D X_a$
$C \to X_a C \mid a$ $Z_1 \to D X_b$.
$D \to X_a Y_1$
$D \to X_b Z_1$
$D \to X_a X_b \mid X_b X_b$

$X_a \to a$ $X_b \to b$

3. $S \to bA \mid aB$
$A \to bAA \mid aS \mid a$
$B \to aBB \mid bS \mid a$.

CNF

1. No ε-production.
2. No unit production.
3. No Useless production.
4. CNF form.

$S \to X_b A \mid X_a B$
$A \to X_b X_1 \mid X_a S \mid a$
$B \to X_a X_2 \mid X_b S \mid a$
$X_1 \to AA$
$X_2 \to BB$
$X_a \to a$
$X_b \to b$

4.

$S \to 1A \mid 0B$

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

$B \to 0BB \mid 1.$

CNF:

1. No ε- production
2. No unit production
3. No useless symbol
4. CNF form.,.

$S \to X_1 A \mid X_0 B$

$A \to X_1 X_2 \mid X_0 S \mid 0$

$B \to X_0 X_3 \mid 1$

$X_0 \to 0$

$X_1 \to 1$

$X_2 \to AA$

$X_3 \to BB$

6.

$S \to abSb \mid a \mid aAb$

$A \to bS \mid aAAb$

CNF:

1, 2, 3 no ε, unit & useless production,

4. CNF Form.,.

$S \to X_1 X_2 \mid a \mid X_a X_4$

$A \to X_b S \mid X_5 X_4$

$X_a \to a$

$X_b \to b$

$X_1 \to X_a X_b$

$X_2 \to S X_b$

$X_3 \to X_a X_4$

$X_4 \to A X_b$

$X_5 \to X_a A$

5.

$S \to a$

$S \to b$

$S \to cSS$

CNF:

1. No ε-production
2. No unit production
3. No Useless production
4. CNF Form.,.

$S \to a$

$S \to b$

$S \to cX$

$X \to SS$

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

7. Design a CFG for the language $L = \{a^{4n} \mid n \geq 1\}$ and convert CFG into CNF form.

$S \to AAAAS \mid AAAA$

$A \to a.$

CNF:

1, 2, 3, no ε, unit & useless production.

$S \to X_1 X_3 \mid X_1 X_1$

$A \to a$

$X_1 \to AA$

$X_3 \to X_1 S$

Greibach normal form (GNF):

The CFG is said to be in GNF if all the productions are of the form..

$$A \rightarrow a\alpha$$

where $a \in T$ and $\alpha \in V^*$

Converting a CFG to GNF:

(*1). Obtain the grammar in CNF.

(*2). Rename the non-terminals to $A_1, A_2, A_3, \ldots$

(*3). Using substitution method obtain the productions to the form

$$A_i \rightarrow A_j \alpha \quad \text{for } i < j$$

where $\alpha \in V^*$

Substitution Method:

$$\left.\begin{aligned} A &\rightarrow x_1 B x_2 \\ B &\rightarrow y_1 | y_2 | \cdots | y_n \end{aligned}\right\} P$$

Then

$$\left.\begin{aligned} A \rightarrow x_1 y_1 x_2 \,|\, x_1 y_2 x_2 \,|\, \cdots \\ \cdots \,|\, x_1 y_n x_2 \end{aligned}\right\} P_1$$

$\boxed{\text{Remove } B \rightarrow y_1 | y_2 | \cdots | y_n}$

(*4). After substitution, if a grammar has left recursion, eliminate left recursion..

Left recursion:

$$A \rightarrow A\alpha_1 \,|\, A\alpha_2 \,|\, A\alpha_3 \cdots \cdots A\alpha_n \,|\, \beta_1 \,|\, \beta_2 \,|\, \beta_3 \cdots \beta_m$$

After removing Left recursion..

$$A \rightarrow \beta_1 A' \,|\, \beta_2 A' \,|\, \beta_3 A' \,|\, \cdots \,|\, \beta_m A'$$

$$A' \rightarrow \alpha_1 A' \,|\, \alpha_2 A' \,|\, \alpha_3 A' \,|\, \cdots \,|\, \alpha_n A' \,|\, \varepsilon$$

Scanned by CamScanner

(4). Step 3 and step 4 can be repeated for any number of times.

(1). Convert the grammar,.

$S \rightarrow AB|BC$
$A \rightarrow aB|bA|a$
$B \rightarrow bB|cC|b$
$C \rightarrow c$ into GNF.

$S \rightarrow AB|BC$ — (1)
$A \rightarrow aB|bA|a$ — (2)
$B \rightarrow bB|cC|b$ — (3)
$C \rightarrow c$ — (4)

(2), (3), & (4) are in GNF.

$S \rightarrow AB|BC$ — (1)

replace (1) with (2) and (3)

$\therefore$ $S \longrightarrow aBB|bAB|aB|bBC|cCC|bC$
$A \longrightarrow aB|bA|a$
$B \longrightarrow bB|cC|b$
$C \rightarrow c$

(2). $S \rightarrow abaSa|aba$

Introduce ,. 2 new productions.,.

$A \rightarrow a$
$B \rightarrow b$

$\therefore$ GNF
$S \rightarrow aBASA|aBA$
$A \rightarrow a$
$B \rightarrow b$

③ $S \rightarrow AB$

$A \rightarrow BS \mid a$

$B \rightarrow SA \mid b$

(*1) Grammar is in CNF

(*2) Rename the variables.,.

$A_1 \rightarrow A_2 A_3$

$A_2 \rightarrow A_3 A_1 \mid a$

$A_3 \rightarrow A_1 A_2 \mid b$.

(*3).

$$\left.\begin{array}{l} A_1 \rightarrow A_2 A_3 \\ A_2 \rightarrow A_3 A_1 \mid a \\ A_3 \rightarrow b \end{array}\right\} \text{can be converted to GNF easily.}$$

$A_3 \rightarrow A_1 A_2$

Replace A_1.

$\therefore A_3 \rightarrow A_2 A_3 A_2$

Replace Light.side A_2

$\therefore A_3 \longrightarrow A_3 A_1 A_3 A_2 \mid a A_3 A_2 \mid b$

(*4). Remove left-Recursion

$A_3 \rightarrow a A_3 A_2 B_3 \mid b B_3 \mid a A_3 A_2 \mid b$ — ③

$B_3 \rightarrow A_1 A_3 A_2 B_3 \mid A_1 A_3 A_2$

ⓐ. Replace A_3 in A_2

$\therefore A_2 \rightarrow a A_3 A_2 B_3 A_1 \mid b B_3 A_1 \mid a A_3 A_2 A_1 \mid b A_1 \mid a$ — ②

ⓑ. Replace

$\therefore A_1$ —

ⓒ. Replace

$\therefore B_3 \rightarrow$

doubt

①, ②,

required GNF

(4) $S \rightarrow$

$A \rightarrow$

$B \rightarrow$

(*1) A & B

(*2) S

S

A

B

(5). $S \rightarrow$

S

A

B

(b). Replace A_2 in A_1.

$\therefore A_1 \rightarrow aA_3A_2B_3A_1A_3 \mid bB_3A_1A_3 \mid aA_3A_2A_1A_3 \mid bA_1A_3 \mid aA_3$ — (1)

(c). Replace A_1 in B_3.

$\therefore B_3 \rightarrow aA_3A_2B_3A_1A_3A_3A_2B_3 \mid bB_3A_1A_3A_3A_2B_3 \mid aA_3A_2A_1A_3A_3A_2B_3 \mid bA_1A_3A_3A_2B_3 \mid aA_3A_3A_2B_3 \mid aA_3A_2B_3A_1A_3A_3A_2 \mid bB_3A_1A_3A_3A_2 \mid aA_3A_2A_1A_3A_3A_2 \mid bA_1A_3A_3A_2 \mid aA_3A_3A_2$ — (4)

doubt

(1), (2), (3) & (4) forms the production rules of the required GNF.

(4) $S \rightarrow AB$

$A \rightarrow aA \mid bB \mid b$

$B \rightarrow b$

(*1) A & B is in GNF

(*2) $S \rightarrow AB$

$S \rightarrow aAB \mid bBB \mid bB$

$A \rightarrow aA \mid bB \mid b$

$B \rightarrow b$ is in GNF.

(5). $S \rightarrow absb \mid aa$

$S \rightarrow aBSB \mid aA$

$A \rightarrow a$

$B \rightarrow b$.

6. $S \to aAS$

doubt

$S \to a$

$A \to SbA$

$B \to SS$

$A \to ba$

$S \to aAS$

$S \to a$

$A \to bA_1$

$A_1 \to a$

$B_1 \to b$

$A \to aASB_1A$

$A \to aB_1A$

$B \to aASS$

$B \to aS$

$A \to SbA$

$\to aASbA$

$\to aASB_1A$

$A \to SbA$

$\to aB_1A$

$B \to SS$

$\to aASS$

7. $S \to AA \mid 0$

$A \to SS \mid 1$

(*1). Grammar is in CNF.

(*2). Rename the variables & Substitution

$A_1 \to A_2A_2 \mid 0$

$A_2 \to A_1A_1 \mid 1.$

$A_2 \to A_2A_2A_1 \mid 0A_1 \mid 1$

(*3). Remove left recursion.

$A_2 \to 0A_1 \mid 1 \mid 0A_1B \mid 1B$ — (2)

$B \to A_2A_1 \mid A_2A_1B$

(*4). A_1

(*5). S

1,

8. Conv

(4). Sub A_2 in A_1

$A_1 \rightarrow 0A_1A_2 \mid 1A_2 \mid 0A_1BA_2 \mid 1BA_2 \mid 0$ — (1)

(5). Sub A_2 in B

$B \rightarrow 0A_1A_1 \mid 1A_1 \mid 0A_1BA_1 \mid 1BA_1 \mid$
$0A_1A_1B \mid 1A_1B \mid 0A_1BA_1B \mid 1BA_1B$ — (3)

(1), (2) and (3) is in GNF.

(8). Convert the following grammar to GNF.

$E \rightarrow E * T \mid T$

$T \rightarrow F + T \mid F$

$F \rightarrow a \mid b \mid (E)$.

Push Down Automata (PDA): (FA + Stack)

PDA is defined by $(Q, \Sigma, \delta, q_0, Z_0, F_0, \Gamma)$

Q : Finite set of states.

Σ : input symbol.

δ : transition function.

q_0 : Initial state.

Z_0 : Bottom of the stack.

F : set of final states. $\Rightarrow$ without final states also we can design PDA.

Γ : Stack alphabet.

PDA

- Deterministic PDA

$$\delta : Q \times \{\Sigma \cup \epsilon\} \times \Gamma \rightarrow Q \times \Gamma^*$$

- Non-deterministic PDA

$$\delta : Q \times \{\Sigma \cup \epsilon\} \times \Gamma \rightarrow 2^{(Q \times \Gamma^*)}$$

Instantaneous Description (ID):

During execution, PDA goes through a sequence of configurations. Each configuration is called instantaneous description (ID) consists of (i). state (ii). input yet to be scanned and (iii) stack content & is represented by a triplet (q, x, γ) where q is the current state, x is the input yet to be read (leftmost symbol of x is the current i/p) and γ is the stack content (leftmost symbol of γ is the current top-of-stack symbol).

Move:

Each move involves a change from one ID to another. The symbol $\vdash$ is used to represent a move.

$(p, ax, A\alpha) \vdash (q, x, \gamma\alpha)$ if $\delta(p, a, A)$ includes (q, γ). [The symbol a is consumed. PDA changes to state q and A is replaced by γ]. $\vdash^*$ represents a sequence of moves.

Language Accepted by a PDA:

PDA can accept a string in 2 ways.

(a). Acceptance by Final State:

An input w is said to be accepted by final state iff $(q_0, w, Z_0) \vdash^* (p, \epsilon, \gamma)$ for some p in F.

Language accepted by a PDA M by final state L(M) is defined as:

$$L(M) = \{w \mid (q_0, w, Z_0) \vdash^* (p, \epsilon, \gamma) \text{ for some } p \text{ in } F\}$$

(b). Acceptance by Empty Stack:

An input w is said to be accepted by a PDA by empty stack iff $(q_0, w, Z_0) \vdash^* (p, \epsilon, \epsilon)$

Language accepted by a PDA M by empty stack N(M) is defined as:

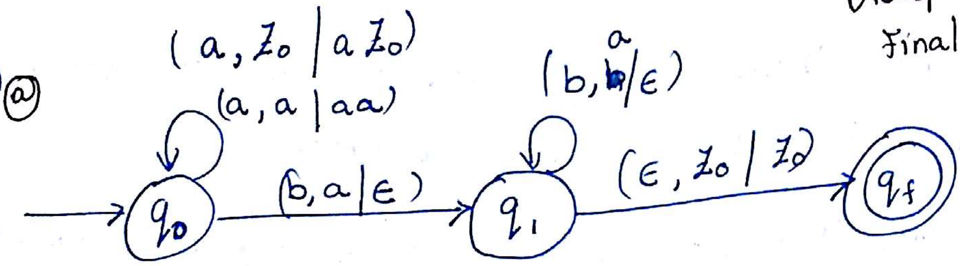
$$N(M) = \{w \mid (q_0, w, Z_0) \vdash^* (q, \epsilon, \epsilon)\}.$$

Note: a & b are equivalent in power.

Scanned by CamScanner

Acceptance by Finalstate.

1 (a)



$a, Z_0 \mid a Z_0$
$a, a \mid aa$ } ⇒ Pushing the elements

$b, a \mid \epsilon$
$b, b \mid \epsilon$ } ⇒ Poping

$\epsilon, Z_0 \mid Z_0$ ⇒

ϵ ⇓ end of string

Z_0 ⇓ top of stack

Equivalent representation of the Automata:

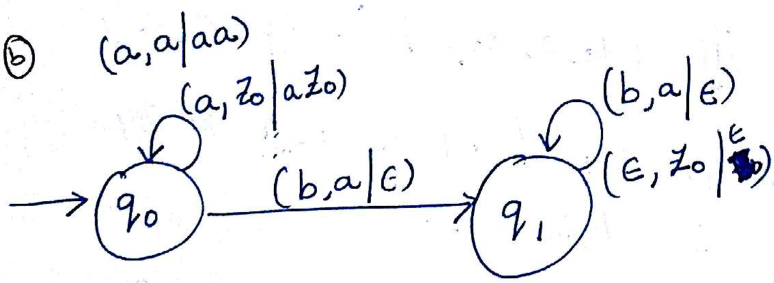
Acceptance by Final state

$$\delta(q_0, a, Z_0) = (q_0, a Z_0)$$
$$\delta(q_0, a, aa) = (q_0, aa)$$
$$\delta(q_0, b, a) = (q_1, \epsilon)$$
$$\delta(q_1, b, a) = (q_1, \epsilon)$$
$$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$$

Acceptance by Emptystack

$$\delta(q_0, a, Z_0) = (q_0, a Z_0)$$
$$\delta(q_0, a, aa) = (q_0, aa)$$
$$\delta(q_0, b, a) = (q_1, \epsilon)$$
$$\delta(q_1, b, a) = (q_1, \epsilon)$$
$$\delta(q_1, \epsilon, Z_0) = (q_1, \cancel{Z_0}\ \epsilon)$$

1 (b)



Acceptance by Empty Stack.

Instantaneous Description :

(i). aabb using (1a)

$(q_0, aabb, Z_0) \vdash (q_0, abb, aZ_0)$

$\vdash (q_0, bb, aaZ_0)$

$\vdash (q_1, b, aZ_0)$

$\cancel{\vdash (q_1, b, Z_0)} \vdash (q_f, \epsilon, Z_0)$ String accepted

$\cancel{\vdash (q_1, \epsilon, Z_0)}$

(ii) aabb using (1b)

$(q_0, aabb, Z_0) \vdash (q_0, abb, aZ_0)$

$\vdash (q_0, bb, aaZ_0)$

$\vdash (q_1, b, aZ_0)$

$\vdash (q_1, \epsilon, Z_0)$ string accepted

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

(iii) aabba using (1a)

$(q_0, aabba, Z_0) \vdash (q_0, abba, aZ_0)$

$\vdash (q_0, bba, aaZ_0)$

$\vdash (q_1, ba, aZ_0)$

$\vdash (q_1, a, Z_0)$ No next move.

(iv) aabba using (1b)

$(q_0, aabba, Z_0) \vdash (q_0, abba, aZ_0)$

$\vdash (q_0, bba, aaZ_0)$

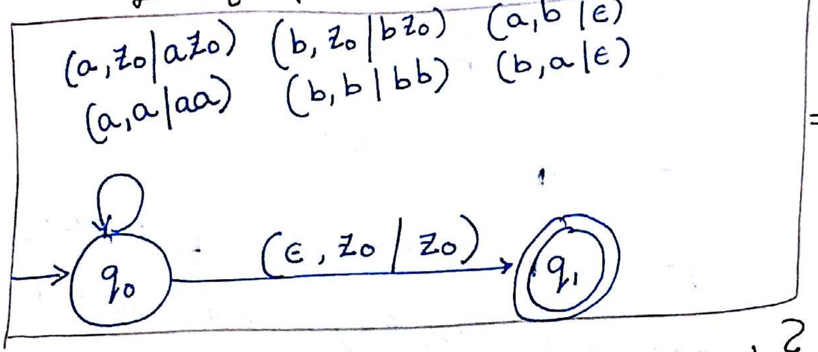
$\vdash (q_1, ba, aZ_0)$

$\vdash (q_1, a, Z_0)$ No next move.

Scanned by CamScanner

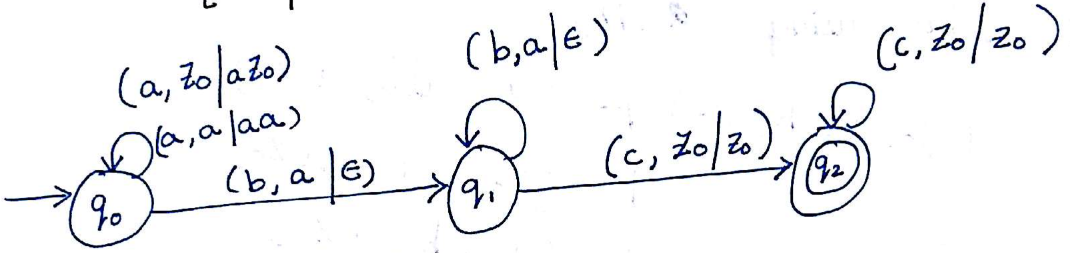
① Design DPDA (or) NDPDA for the foll. languages.

① $L = \{ w \mid n_a(w) = n_b(w) \}$

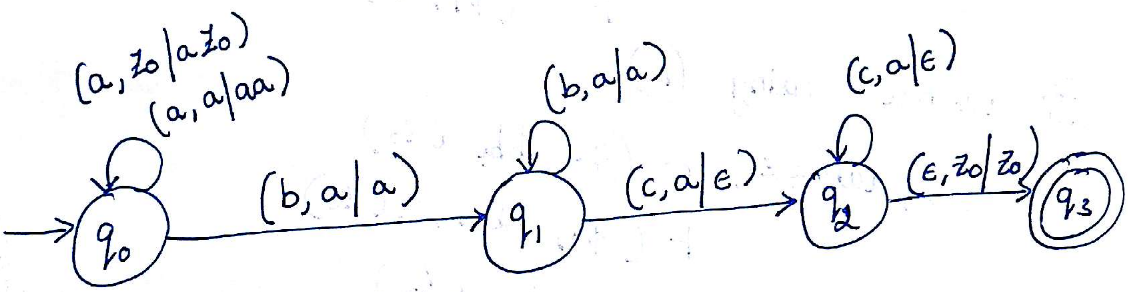


⇒ Non-deterministic PDA.

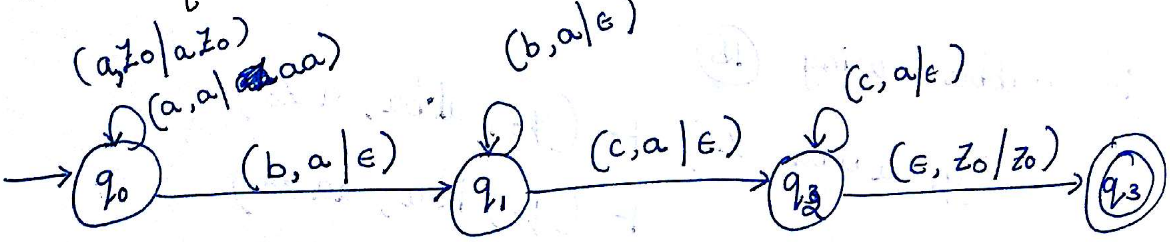
② $L = \{ w \mid a^n b^n c^m, n, m \geq 1 \}$



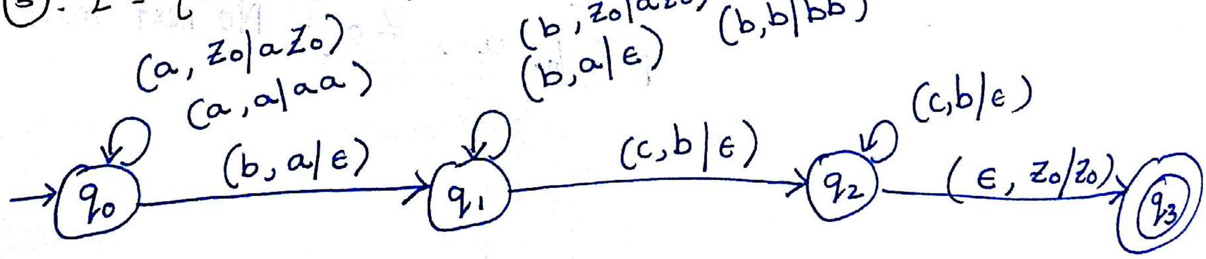
③ $L = \{ a^n b^m c^n \mid n, m \geq 1 \}$



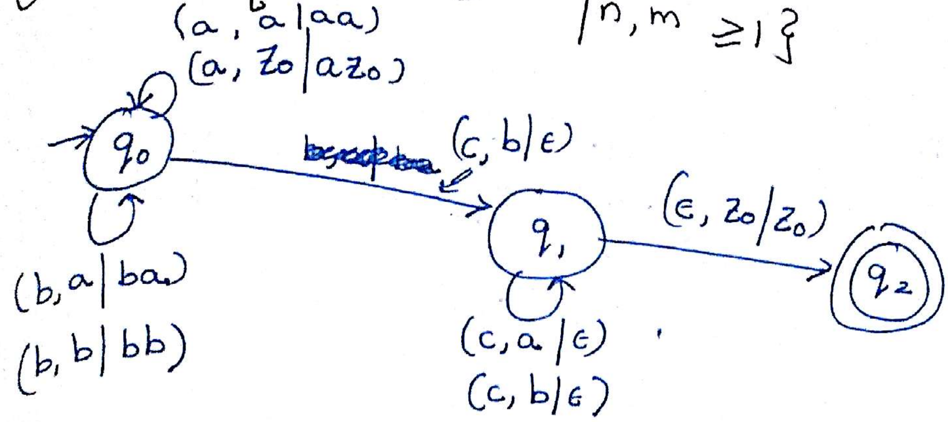
④ $L = \{ a^{m+n} b^m c^n \mid m, n \geq 1 \}$



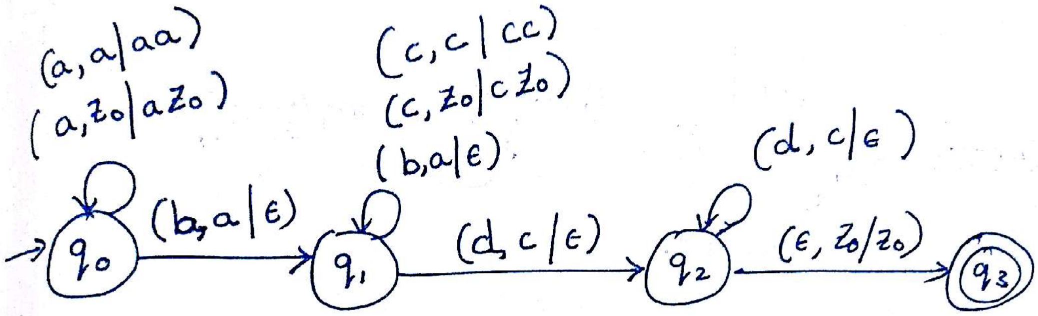
⑤ $L = \{ a^n b^{m+n} c^m \mid n, m \geq 1 \}$



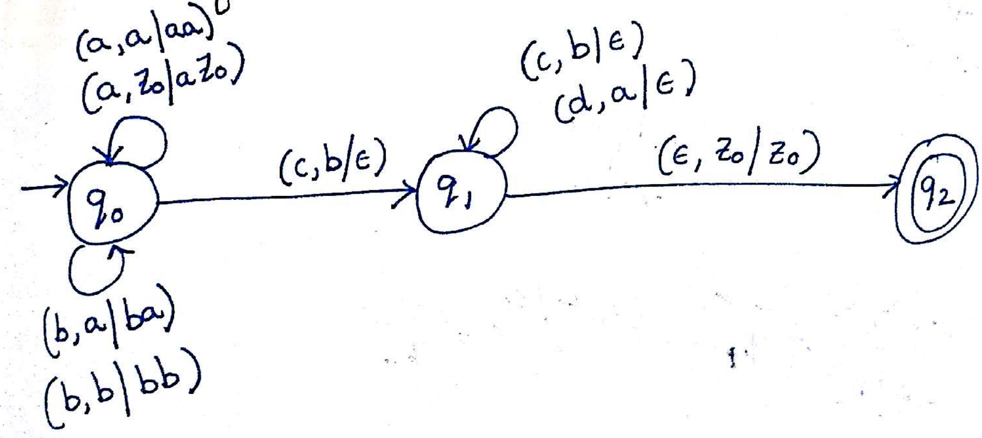
6. $L = \{a^n b^m c^{n+m} \mid n, m \geq 1\}$



7. $L = \{a^n b^n c^m d^m \mid n, m \geq 1\}$

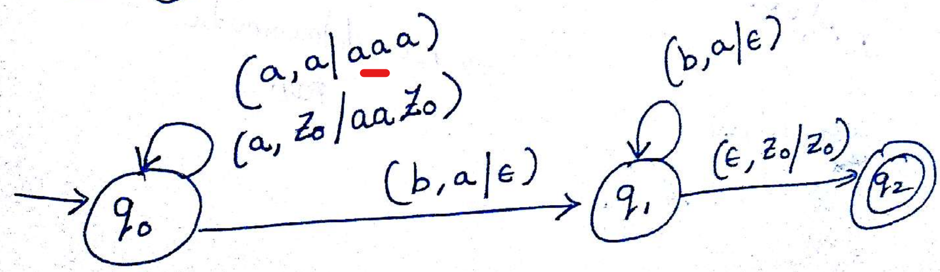


8. $L = \{a^n b^m c^m d^n \mid n, m \geq 1\}$



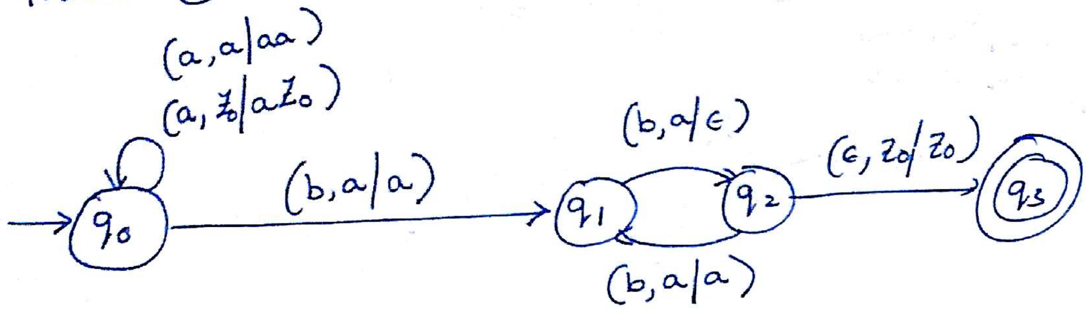
9. $a^n b^{2n} \mid n \geq 1$

Solution ①:

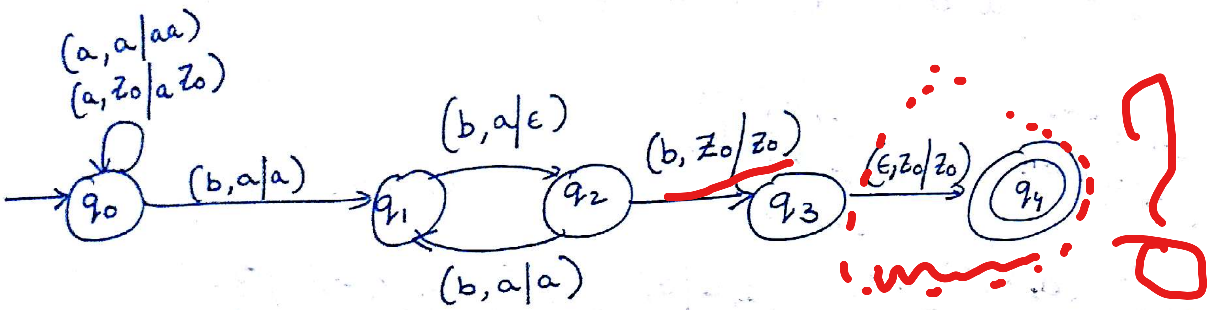


Scanned by CamScanner

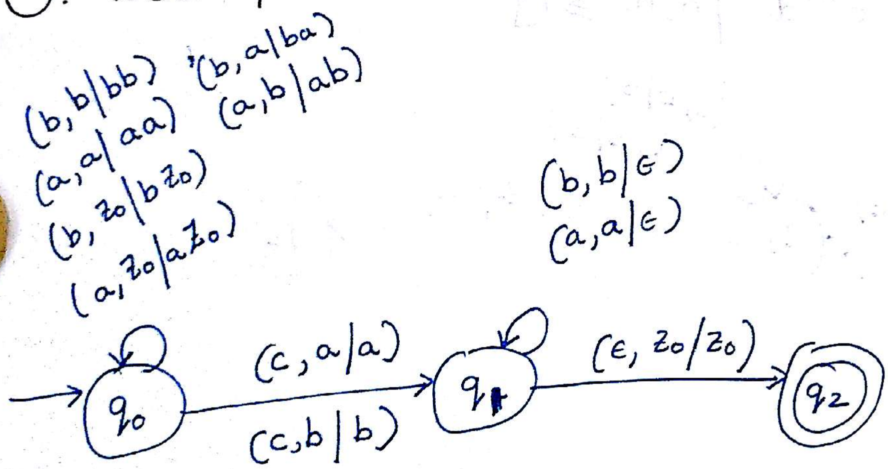
Solution ②:



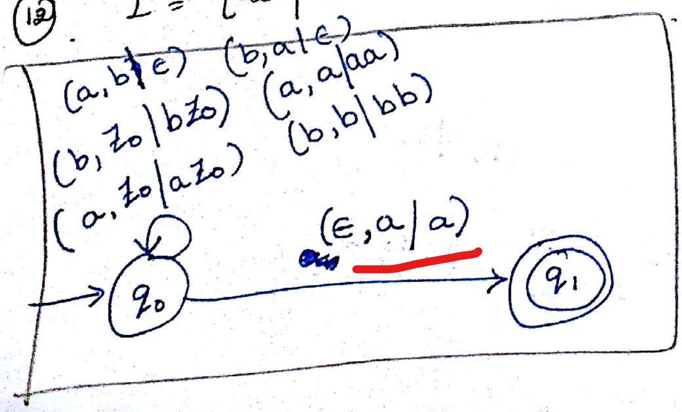
⑩. $a^n b^{2n+1}$



⑪. $wcw^R \mid w \in (a,b)^+$

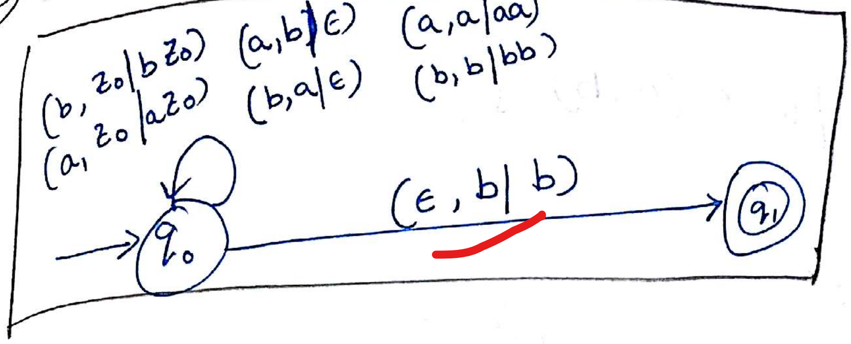


⑫. $L = \{w \mid w \in (a,b)^*$ & $n_a(w) > n_b(w)\}$.



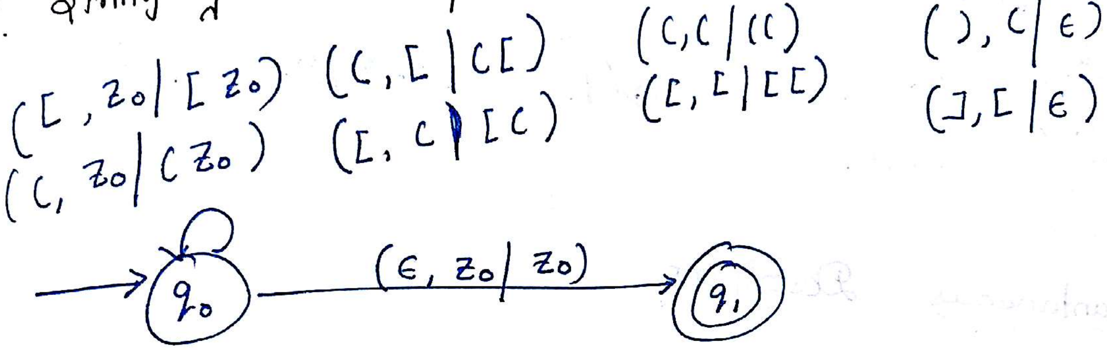
=> Non-deterministic PDA.

13\. $L = \{ w \mid w \in (a,b)^* \;\&\; n_a(w) < n_b(w) \}$



⇒ Non deterministic PDA

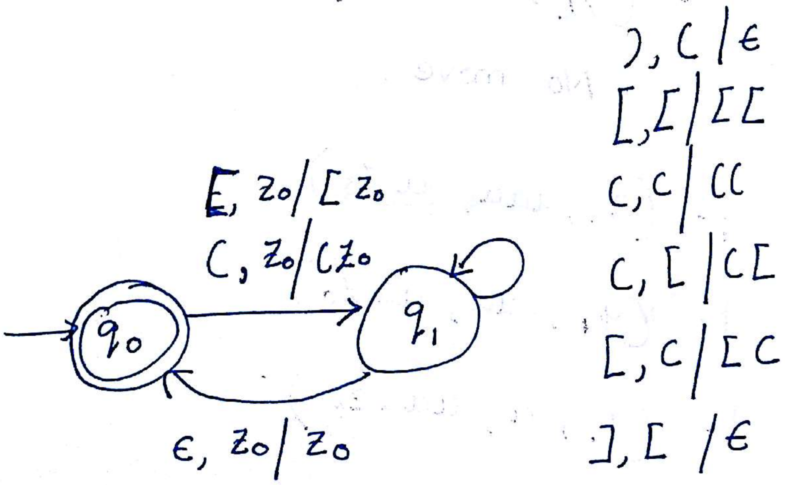
14\. String of balanced parentheses. (,), [,].



15\. String of balanced parentheses.

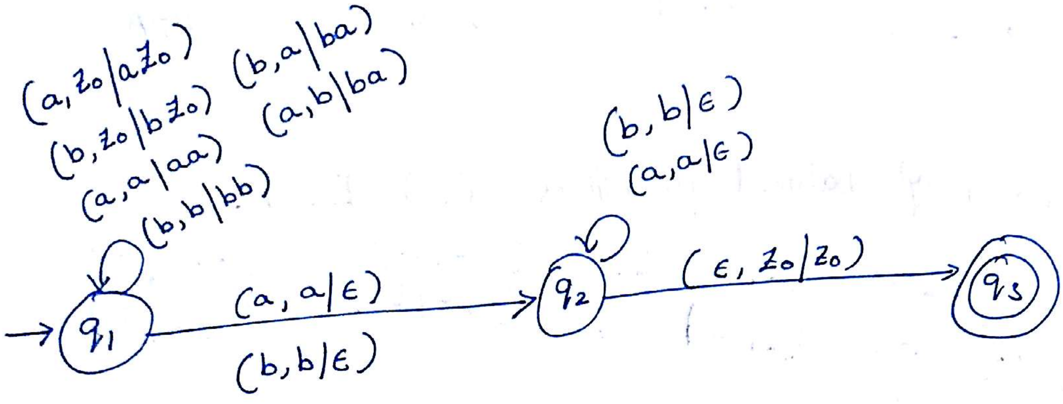
Valid strings: ε , [()()([ ])] , [ ][ ]()

Invalid strings: [ )()[ ],)(, [ ) .



② Design NPDA (or) DPDA for the foll. languages.

① $L = \{ww^R \mid w \in (a,b)^+\}$.



Instantaneous Description:

aaaa

↓

$(q_1, aaaa, z_0) \vdash (q_1, aaa, az_0)$

$\vdash (q_2, aa, z_0)$

No move.

$(q_1, aaaa, z_0) \vdash (q_1, aaa, az_0)$

$\vdash (q_1, aa, aaz_0)$

$\vdash (q_1, a, aaaz_0)$

$\vdash (q_1, \epsilon, aaaaz_0)$

No move.

$(q_1, aaaa, z_0) \vdash (q_1, aaa, az_0)$

$\vdash (q_1, aa, aaz_0)$

$\vdash (q_1, a, aaaz_0)$

$\vdash (q_1, \epsilon, aaz_0)$

No move

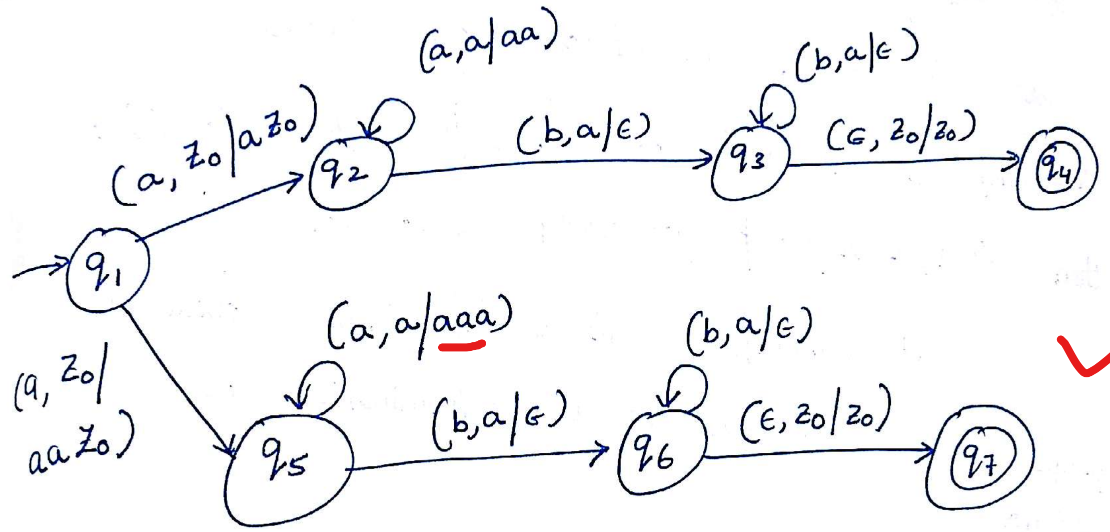
$(q_1, aaaa, Z_0) \vdash (q_1, aaa, aZ_0)$

$\vdash (q_1, aa, aaZ_0)$

$\vdash (q_2, a, aZ_0)$

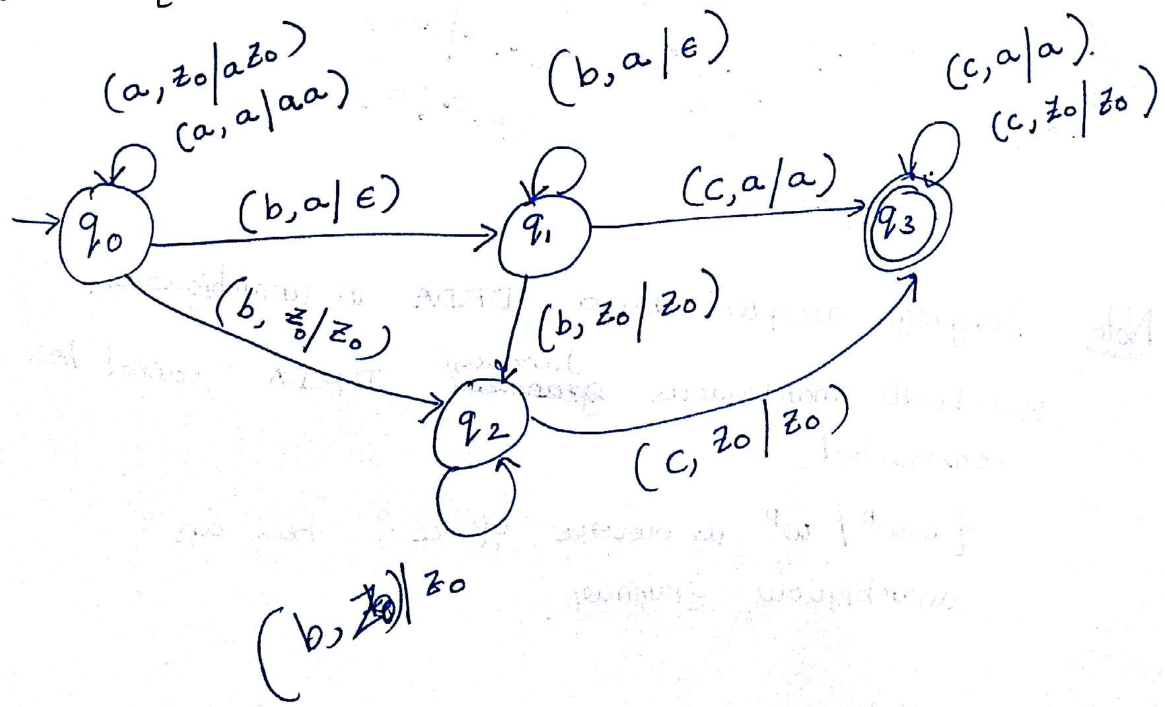
$\vdash (q_2, \epsilon, Z_0)$ String Accepted

② $L = \{a^n b^n \mid n \geq 1\} \cup \{a^n b^{2n} \mid n \geq 1\}$



~~⑫. L = {w | w ∈ (a+b)* and~~

③. $L = \{a^i b^j c^k \mid i \neq j\}$ $i \geq 0, j \geq 1, k \geq 1.$



Deterministic PDA:

Let $M = (Q, \Sigma, \delta, q_0, F, Z_0, \Gamma)$ M is deterministic if there is no configuration for which M has a choice of more than one move.

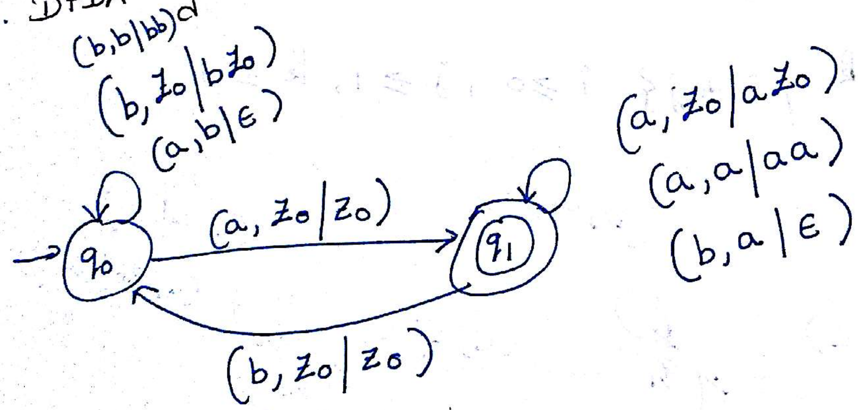
In other words, M is deterministic if it satisfies both the following conditions:

(i). For any $q \in Q$, $a \in \Sigma \cup \{\epsilon\}$ and $X \in \Gamma$, the set $\delta(q, a, X)$ has atmost one element.

(ii). For any $q \in Q$ and $X \in \Gamma$, if $\delta(q, \epsilon, X) \neq \phi$, then $\delta(q, a, X) = \phi$ for every $a \in \Sigma$.

A language L is a deterministic context-free language (DCFL) if there is a deterministic PDA (DPDA) accepting L.

Q. DPDA for $L = \{x \in \{a,b\}^* \mid n_a(x) > n_b(x)\}$.



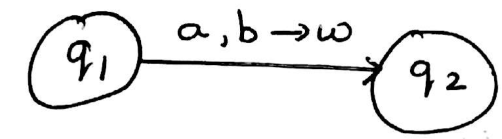
Note: Language accepted by a DPDA is unambiguous.

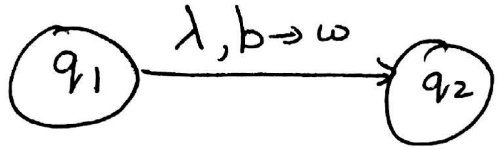
But for All unambiguous ~~grammar~~ Language DPDA cannot be constructed.

$\{ww^R \mid w^R$ is reverse of $w\}$ has an unambiguous grammar.

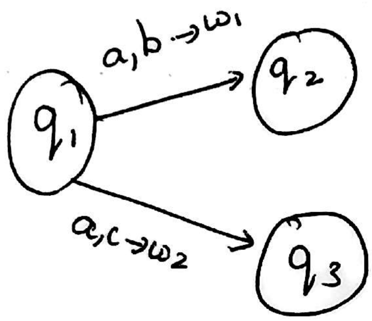
Allowed Transitions in DPDA :

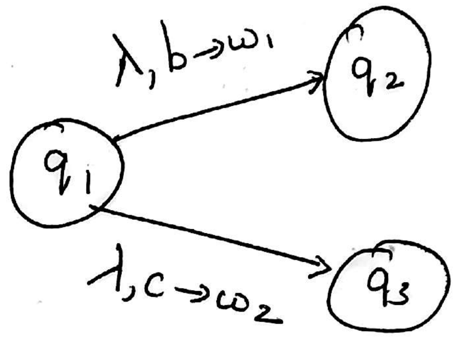
*



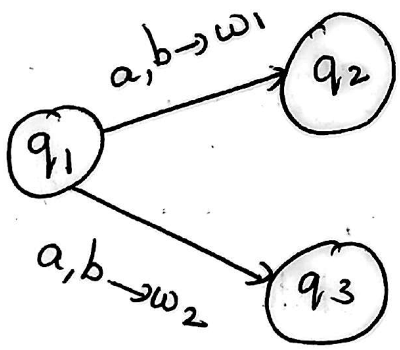


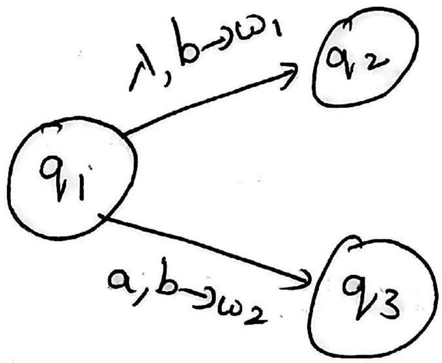
*





Not - Allowed Transitions in DPDA :





CFG to PDA:

Conversion of CFG to PDA

1. Convert the grammar into GNF.

2. For $q_0 \rightarrow$ start state

   $Z_0 \rightarrow$ top of stack

   without consuming any i/p, push the start symbol S onto the stack and change the state to q_1. The transition for this can be

$$\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$$

3. For each production of the form

$$A \rightarrow a\alpha$$

introduce the transition

$$\delta(q_1, a, A) = (q_1, \alpha)$$

$A \rightarrow a$ introduce the tran

$$\delta(q_1, a, A) = (q_1, \epsilon).$$

4. At state q_1, without consuming any i/p, change the state to q_f which is an accepting state.

   The transition will be

$$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0).$$

1. $S \rightarrow aABC$

   $A \rightarrow aB \mid a$

   $B \rightarrow bA \mid b$

   $C \rightarrow a.$

   1. $\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$
   2. $\delta(q_1, a, S) = (q_1, ABC)$
   3. $\delta(q_1, a, A) = (q_1, B)$
   4. $\delta(q_1, a, A) = (q_1, \epsilon)$

⑤. $\delta(q_1, b, B) = (q_1, A)$

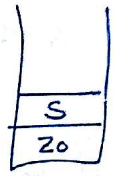
⑥. $\delta(q_1, b, B) = (q_1, \epsilon)$

⑦. $\delta(q_1, a, C) = (q_1, \epsilon)$

⑧. $\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$.

<u>aaba</u>

$$\begin{aligned} S &\Rightarrow aABC \\ &\Rightarrow aaBC \\ &\Rightarrow aabC \\ &\Rightarrow aaba \end{aligned}$$



From PDA

$$\begin{aligned} (q_0, aaba, Z_0) &\vdash (q_1, aaba, SZ_0) \\ &\vdash (q_1, aba, ABCZ_0) \\ &\vdash (q_1, ba, BCZ_0) \\ &\vdash (q_1, a, CZ_0) \\ &\vdash (q_1, \epsilon, Z_0) \\ &\vdash (q_f, \epsilon, Z_0). \end{aligned}$$

ⓓ. $L = \{a^n b^n \mid n \geq 1\}$

$S \to aSb$

$S \to ab$

$S \to aSB$

$S \to aB$

$B \to b$

$\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$

$\delta(q_1, a, S) = (q_1, SB)$

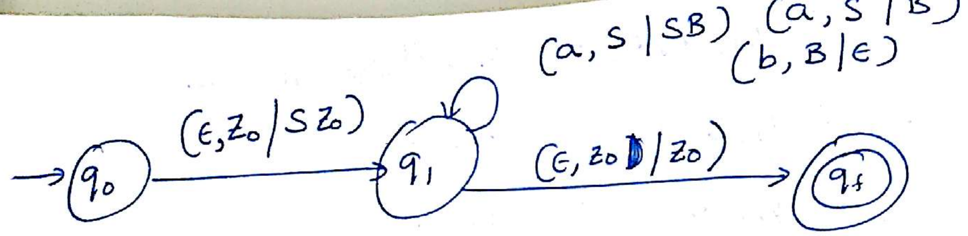
$\delta(q_1, a, S) = (q_1, B)$

$\delta(q_1, b, B) = (q_1, \epsilon)$

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

$M = (\{q_0, q_1, q_f\}, \{a, b\}, \delta, q_0, q_f, Z_0, \{S, A, B, Z_0\})$.

Scanned by CamScanner



③. $S \rightarrow 0AA$

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

$\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$

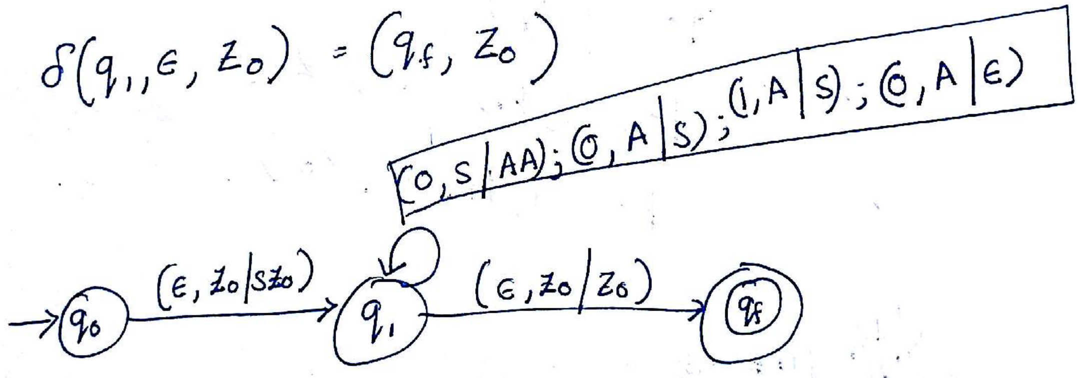
$\delta(q_1, 0, S) = (q_1, AA)$

$\delta(q_1, 0, A) = (q_1, S)$

$\delta(q_1, 1, A) = (q_1, S)$

$\delta(q_1, 0, A) = (q_1, \epsilon)$

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



$010000 \in L(G)$.

$(q_0, 010000, Z_0) \vdash (q_1, \underline{0}10000, \underline{S}Z_0)$

$\vdash (q_1, \underline{1}0000, \underline{A}AZ_0)$

$\vdash (q_1, \underline{0}000, \underline{S}AZ_0)$

$\vdash (q_1, \underline{0}00, \underline{A}AAZ_0)$

$\vdash (q_1, \underline{0}0, \underline{A}AZ_0)$

$\vdash (q_1, \underline{0}, \underline{A}Z_0)$

$\vdash (q_1, \epsilon, Z_0)$

$\vdash (q_f, \epsilon, Z_0)$.

String Accepted.

CFG to PDA: (without considering GNF).

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

$$P = (\{q\}, T, \delta, q, \phi, S, V \cup T)$$

No final state.

This type of conversion will design PDA with empty stack.

Conversion from CFG to PDA:

①. For variable A:

$$\delta(q, \epsilon, A) = \{(q, \beta) \mid A \to \beta \text{ is a production of } P\}$$

②. Terminal a:

$$\delta(q, a, a) = \{q, \epsilon\}.$$

①. $I \to a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

$E \to I \mid E*E \mid E+E \mid (E).$

①. For variable E & I

$$\delta(q, \epsilon, E) = \{(q, E*E), (q, E+E), (q, (E)), (q, I)\}$$

$$\delta(q, \epsilon, I) = \{(q, a), (q, b), (q, Ia), (q, Ib), (q, I0), (q, I1)\}$$

②. Terminal a, b, 0, 1, *, +, (,)

$\delta(q, a, a) = \{q, \epsilon\}$

$\delta(q, b, b) = \{q, \epsilon\}$

$\delta(q, 0, 0) = \{q, \epsilon\}$

$\delta(q, 1, 1) = \{q, \epsilon\}$

$\delta(q, *, *) = \{q, \epsilon\}$

$\delta(q, +, +) = \{q, \epsilon\}$

$\delta(q, (, () = \{q, \epsilon\}$

$\delta(q,),)) = \{q, \epsilon\}$

(2). $S \longrightarrow 0S1 \mid A$

$A \longrightarrow 1A0 \mid S \mid \epsilon$

Variable S & A:

$\delta(q, \epsilon, S) = \{(q, 0S1), (q, A)\}$

$\delta(q, \epsilon, A) = \{(q, 1A0), (q, S), (q, \epsilon)\}$

Variable 0 & 1:

$\delta(q, 0, 0) = \{q, \epsilon\}$

$\delta(q, 1, 1) = \{q, \epsilon\}$

Tracing: 0101

$\delta(q, 0101, S) \vdash \delta(q, 0101, 0S1)$

$\vdash \delta(q, 101, S1)$

$\vdash \delta(q, 101, A1)$

$\vdash \delta(q, 101, 1A01)$

$\vdash \delta(q, 01, A01)$

$\vdash \delta(q, 01, 01)$

$\vdash \delta(q, 1, 1)$

$\vdash \delta(q, \epsilon, \epsilon)$.

String is accepted.

Construction of CFG Equivalent of a PDA:

Let $P = (Q, \Sigma, \delta, q_0, \phi, Z_0, \Gamma)$ be a PDA that accepts a language L by empty stack.

CFG $G = (V, T, P, S)$ that generates L can be constructed using the following rules:

1. For every $q \in Q$, add a production $S \rightarrow [q_0 Z q]$ in P. Thus, if there are n states in a PDA P, then there will be 'n' new productions added in P.

2. For every $q, r \in Q$, $a \in \{\Sigma \cup \epsilon\}$, $X \in \Gamma$, if $\delta(q, a, X) = (r, \epsilon)$ then add a production

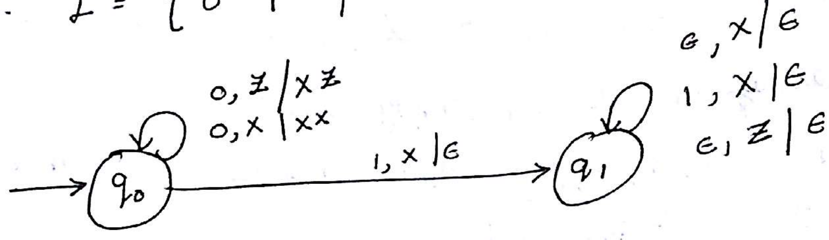
$$[q X r] \rightarrow a \text{ in } P.$$

3. For every $q, r \in Q$, $a \in \{\Sigma \cup \epsilon\}$, $X \in \Gamma$ and $x \geq 1$, if $\delta(q, a, X) = (r, X_1 X_2 \ldots X_k)$ where for every choice of $X_1, X_2, \ldots X_k \in \Gamma$, then for every choice of $q_1, q_2, \ldots q_k \in Q$ add the production

$$[q X q_k] \rightarrow a [r X_1 q_1][q_1 X_2 q_2] \ldots [q_{k-1} X_k q_k]$$

in P.

① $L = \{0^i 1^j \mid i \geq j \geq 1\}$



Using Rule ①:

$$S \longrightarrow [q_0 Z q_0] \mid [q_0 Z q_1]$$

Using Rule ②:

$\delta(q_0, 1, X) = (q_1, \epsilon) \Rightarrow [q_0 X q_1] \longrightarrow 1$

$\delta(q_1, \epsilon, X) = (q_1, \epsilon) \Rightarrow [q_1 X q_1] \longrightarrow \epsilon$

$\delta(q_1, 1, X) = (q_1, \epsilon) \Rightarrow [q_1 X q_1] \longrightarrow 1$

$\delta(q_1, \epsilon, Z) = (q_1, \epsilon) \Rightarrow [q_1 Z q_1] \longrightarrow \epsilon$

Using Rule ③:

Ⅰ $\delta(q_0, 0, Z) = (q_0, XZ)$

$[q_0 Z q_0] \longrightarrow 0 [q_0 X q_0][q_0 Z q_0]$

$\longrightarrow 0 [q_0 X q_1][q_1 Z q_0]$

$[q_0 Z q_1] \longrightarrow 0 [q_0 X q_1][q_1 Z q_1]$

$\longrightarrow 0 [q_0 X q_0][q_0 Z q_1]$

Ⅱ $\delta(q_0, 0, X) = (q_0, XX)$

$[q_0 X q_0] \longrightarrow 0 [q_0 X q_0][q_0 X q_0]$

$\longrightarrow 0 [q_0 X q_1][q_1 X q_0]$

$[q_0 X q_1] \longrightarrow 0 [q_0 X q_1][q_1 X q_1]$

$\longrightarrow 0 [q_0 X q_0][q_0 X q_1]$

∴ The required Grammar G . (V, T, P, S) will be

$V = \{ S, [q_0 Z q_0], [q_0 Z q_1], [q_1 Z q_0], [q_1 Z q_1]$
$[q_0 X q_0], [q_0 X q_1], [q_1 X q_0], [q_1 X q_1] \}$

$T = \{0, 1\}$

$P = \{$

$S \rightarrow [q_0 Z q_1] \mid [q_0 Z q_0]$ — ①, ②

$[q_0 X q_1] \rightarrow 1$ — ③

$[q_1 X q_1] \rightarrow 1$ — ④

$[q_1 X q_1] \rightarrow \varepsilon$ — ⑤

$[q_1 Z q_1] \rightarrow \varepsilon$ — ⑥

$[q_0 Z q_0] \rightarrow 0 [q_0 X q_0] [q_0 Z q_0] \mid$ — ⑦

$\rightarrow 0 [q_0 X q_1] [q_1 Z q_0]$ — ⑧

$[q_0 Z q_1] \rightarrow 0 [q_0 X q_1] [q_1 Z q_1] \mid 0 [q_0 X q_0] [q_0 Z q_1]$ — ⑨, ⑩

$[q_0 X q_0] \rightarrow 0 [q_0 X q_0] [q_0 X q_0] \mid 0 [q_0 X q_1] [q_1 X q_0]$ — ⑪, ⑫

$[q_0 X q_1] \rightarrow 0 [q_0 X q_1] [q_1 X q_1] \mid 0 [q_0 X q_0] [q_0 X q_1]$ — ⑬, ⑭

$\}$

Start symbol S.

~~Tracing~~ Generating:

00011

$S \rightarrow [q_0 Z q_1]$ ①

$\rightarrow 0 [q_0 X q_1] [q_1 Z q_1]$ ⑨

$\rightarrow 0\, 0 [q_0 X q_1] [q_1 X q_1] [q_1 Z q_1]$ ⑬

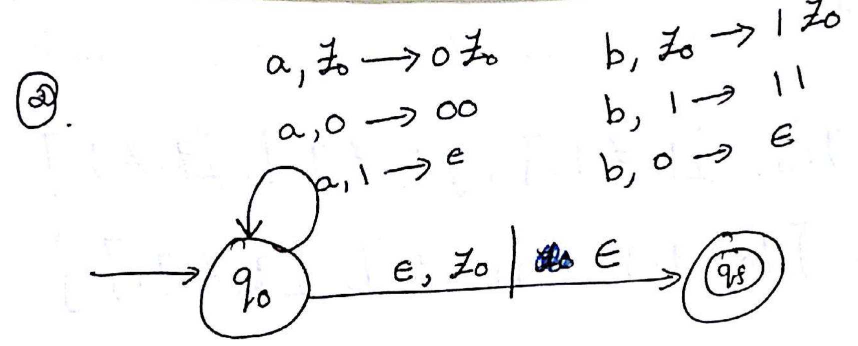
$\rightarrow 000 [q_0 X q_1] [q_1 X q_1] [q_1 Z q_1]$ ⑬

$\rightarrow 000\, 1 [q_1 X q_1] [q_1 Z q_1]$ ③

$\rightarrow 00011 [q_1 Z q_1]$ ④

$\rightarrow 00011\, \varepsilon$ ⑥

(2).



Using Rule ①:

$$S \to [q_0 Z q_0] \quad -① \;\mid\; [q_0 Z q_f] \quad -②$$

Using Rule ②:

$$\delta(q_0, a, 1) = (q_0, \epsilon) \Rightarrow [q_0 \; 1 \; q_0] \to a \quad -③$$

$$\delta(q_0, b, 0) = (q_0, \epsilon) \Rightarrow [q_0 \; 0 \; q_0] \to b \quad -④$$

$$\delta(q_0, \epsilon, Z_0) = (q_f, \epsilon) \Rightarrow [q_0 \; Z_0 \; q_f] \to \epsilon \quad -⑤$$

Using Rule ③:

(I). $\delta(q_0, a, Z_0) = (q_0, 0Z_0)$

$$[q_0 Z_0 q_0] \to a[q_0 0 q_0][q_0 Z_0 q_0] \mid \quad -⑥$$

$$a[q_0 0 q_f][q_f Z_0 q_0] \quad -⑦$$

$$[q_0 Z_0 q_f] \to a[q_0 0 q_0][q_0 Z_0 q_f] \mid \quad -⑧$$

$$a[q_0 0 q_f][q_f Z_0 q_f] \quad -⑨$$

(II). $\delta(q_0, b, Z_0) = (q_0, 1Z_0)$

$$[q_0 Z_0 q_0] \to b[q_0 1 q_0][q_0 Z_0 q_0] \mid \quad -⑩$$

$$b[q_0 1 q_f][q_f Z_0 q_f] \quad -⑪$$

$$[q_0 Z_0 q_f] \to b[q_0 1 q_0][q_0 Z_0 q_f] \mid \quad -⑫$$

$$b[q_0 1 q_f][q_f Z_0 q_f] \quad -⑬$$

(iii). $\delta(q_0, a, 0) = (q_0, 00)$

$[q_0\ 0\ q_0] \rightarrow a[q_0\ 0\ q_0][q_0\ 0\ q_0]\ |$ — (14)

$\qquad a[q_0\ 0\ q_f][q_f\ 0\ q_f]$ — (15)

$[q_0\ 0\ q_f] \rightarrow a[q_0\ 0\ q_0][q_0\ 0\ q_0]\ |$ — (16)

$\qquad a[q_0\ 0\ q_f][q_f\ 0\ q_f]$ — (17)

(iv). $\delta(q_0, b, 1) = (q_0, 11)$

$[q_0\ 1\ q_0] \rightarrow b[q_0\ 1\ q_0][q_0\ 1\ q_0]\ |$ — (18)

$\qquad b[q_0\ 1\ q_f][q_f\ 1\ q_f]$ — (19)

$[q_0\ 1\ q_f] \rightarrow b[q_0\ 1\ q_0][q_0\ 1\ q_f]\ |$ — (20)

$\qquad b[q_0\ 1\ q_f][q_f\ 1\ q_f]$. — (21)

Generating:

abab

$S \longrightarrow [q_0\ Z\ q_f]$ (2)

$\longrightarrow a[q_0\ 0\ q_0][q_0\ Z_0\ q_f]$ ~~(7)~~ ~~(8)~~ (8)

$\rightarrow ab[q_0\ Z_0\ q_f]$ (4)

$\rightarrow aba[q_0\ 0\ q_0][q_0\ Z_0\ q_f]$ (8)

$\rightarrow abab[q_0\ Z_0\ q_f]$ (4)

$\rightarrow abab\ \varepsilon$ (5).

③. $M = (\{q_0, q_1\}, \{a, b\}, \{A, B, Z\}, \delta, q_0, Z)$

where δ is defined as.,.

$\delta(q_0, a, Z) = (q_0, AZ)$

$\delta(q_0, b, Z) = (q_0, BZ)$

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

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

$\delta(q_0, a, B) = \{(q_0, AB)\}$

$\delta(q_0, b, A) = \{(q_0, BA)\}$

$\delta(q_1, a, A) = (q_1, \epsilon)$

$\delta(q_1, b, B) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, Z) = (q_1, \epsilon)$.

Solution:

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

$V = \{S, [q_0 Z q_0], [q_0 A q_0], [q_0 B q_0], [q_1 Z q_0],$
$[q_1 B q_0], [q_1 A q_0], [q_0 Z q_1], [q_0 A q_1],$
$[q_0 B q_1], [q_1 A q_1], [q_1 B q_1], [q_1 Z q_1]\}$

$T = \{a, b\}$

$P =$

$\{$

$S \longrightarrow [q_0 Z q_0] \mid [q_0 Z q_1]$

$[q_0 Z q_0] \longrightarrow a [q_0 A q_0][q_0 Z q_0]$

$\rightarrow a [q_0 A q_1][q_1 Z q_0]$

$[q_0 Z q_1] \rightarrow a [q_0 A q_0][q_0 Z q_1]$

$\rightarrow a [q_0 A q_1][q_1 Z q_1]$

AZ

$[q_0 A q_0] \rightarrow a [q_0 A q_0][q_0 A q_0]$

$\rightarrow a [q_0 A q_1][q_1 A q_0]$

$[q_0 A q_1] \rightarrow a [q_0 A q_0][q_0 A q_1]$

$\rightarrow a [q_0 A q_1][q_1 A q_1]$

AA

$[q_0 Z q_0] \rightarrow b [q_0 B q_0][q_0 Z q_0]$

$\rightarrow b [q_0 B q_1][q_1 Z q_0]$

$[q_0 Z q_1] \rightarrow b [q_0 B q_0][q_0 Z q_1]$

$\rightarrow b [q_0 B q_1][q_1 Z q_1]$

BZ

$[q_0, B q_0] \rightarrow b [q_0 B q_0][q_0 B q_0]$

$\rightarrow b [q_0 B q_1][q_1 B q_0]$

$[q_0, B q_1] \rightarrow b [q_0 B q_0][q_0 B q_1]$

$\rightarrow b [q_0 B q_1][q_1 B q_1]$

BB

$[q_0 B q_0] \rightarrow a [q_0 A q_0][q_0 B q_0]$

$\rightarrow a [q_0 A q_1][q_1 B q_0]$

$[q_0 B q_1] \rightarrow a [q_0 A q_0][q_0 B q_1]$

$\rightarrow a [q_0 A q_1][q_1 B q_1]$

AB

$[q_0 A q_0] \rightarrow b [q_0 B q_0][q_0 A q_0]$

$\rightarrow b [q_0 B q_1][q_1 A q_0]$

$[q_0 A q_1] \rightarrow b [q_0 B q_0][q_0 A q_1]$

$\rightarrow b [q_0 B q_1][q_1 A q_1]$

BA

$[q_0 A q_1] \rightarrow a$ $\quad$ $[q_1 A q_1] \rightarrow a$ $\quad$ $[q_1 Z q_1] \rightarrow \epsilon$

$[q_0 B q_1] \rightarrow b$ $\quad$ $[q_1 B q_1] \rightarrow b$

Scanned by CamScanner

Decision Properties of the Regular Language:

* Membership Problem
* Emptiness of the Language
* Finiteness of the Language

Ex for membership problem:

Given a regular expression r and a string x, does x belong to the language corresponding to r?.

Ex for Emptiness of the language:

Given a finite automaton M, is there a string that it accepts? (Alternatively, given an FA M, is $L(M)=\phi$)?

Ex for finiteness of the language:

Given an FA M, is $L(M)$ finite?.

①. Given 2 FAs M_1 and M_2, is $L(M_1)$ a subset of $L(M_2)$?

Find $L(M_1) - L(M_2)$

$L(M_1) \subseteq L(M_2)$ if & only if $L(M_1)-L(M_2)=\phi$.

②. Given 2 FAs M_1 and M_2, are there any strings that are accepted by neither?

Find M_1' and M_2' accepting $L(M_1)'$ & $L(M_2)'$ and then find $L(M_1') \cap L(M_2')$

(3). Given an FA M accepting a languge L, and given 2 strings x and y are x and y distinguishable with respect to L?

x & y are distinguishable with respect to L if and only if $\delta_1^*(q_1, x) \neq \delta_1^*(q_1, y)$

For each statement below, decide whether it is true or false. If it is true, prove it. If not, give a counterexample. All parts refer to languages over the alphabet $\{0, 1\}$.

(a). If $L_1 \subseteq L_2$ and L_1 is not regular, then L_2 is not regular. $L_1 - L_2 = \phi$

False. Σ^* has a nonregular subset.

(b). If $L_1 \subseteq L_2$ and L_2 is not regular, then L_1 is not regular.

False. Non-regular language have finite subsets and finite languages are regular.

(c). If L_1 and L_2 are non-regular, then $L_1 \cup L_2$ is non-regular.

False. The union of any language and its complement is Σ^*, which is regular.

(d). If L_1 and L_2 are nonregular, then $L_1 \cap L_2$ is non-regular.

False. The intersection of a nonregular language and its complement is empty and the empty language is regular.

(e). If L is non-regular, then L' is nonregular.

True. The complement of a regular lang. is regu.

(f). If L_1 is regular and L_2 is nonregular, then $L_1 \cup L_2$ is nonregular.

False. L_2 could be a subset of L_1.

(g). If L_1 is regular, L_2 is nonregular, and $L_1 \cap L_2$ is regular, then $L_1 \cup L_2$ is nonregular.

True. $L_1 \cap L_2$ is reg. $L_1 - (L_1 \cap L_2) = L_1 - L_2$

$L_2 = (L_1 \cup L_2) - (L_1 - L_2)$ $\therefore L_1 \cup L_2$ reg. L_2 is also reg.

(h). If L_1 is regular, L_2 is non-regular, and $L_1 \cap L_2$ is non-regular then $L_1 \cup L_2$ is non-regular.

False L_1 could be Σ^*

(i). If $L_1, L_2, L_3, \ldots$ are all regular, then $\bigcup_{n=1}^{\infty} L_n$ is regular.

False.

$\{0^n 1^n \mid n \geq 0\} = \{\Lambda\} \cup \{01\} \cup \{0011\} \cup \ldots$

(j). If $L_1, L_2, L_3, \ldots$ are all non-regular and $L_i \subseteq L_{i+1}$ for each i, then $\bigcup_{n=1}^{\infty} L_n$ is nonregular.

False.

For each $k \geq 1$

let $S_k = \{0^{2^k i} 1^{2^k i} \mid i \geq 0\}$

$S_1 = \{\Lambda, 0^2 1^2, 0^4 1^4, 0^6 1^6, \ldots\}$

$S_2 = \{\Lambda, 0^4 1^4, 0^8 1^8, \ldots\}$

$S_{k+1} \subseteq S_k$ for each k.

The set $\bigcap_{k=0}^{\infty} S_k$ is $\{\Lambda\}$ which is regular but it is easy to show that for every k S_k is nonregular.

Now let $L_k = S_k'$. It follows that L_k is non-regular and $L_k \subseteq L_{k+1}$ but

$\bigcup_{k=1}^{\infty} L_k = \left(\bigcap_{k=1}^{\infty} S_k\right)' = \Sigma^* - \{\Lambda\}$ and this set is regular.

CLOSURE PROPERTIES OF REGULAR LANGUAGES

(*1) The union of two regular languages is regular.
(*2) The intersection of two regular languages is regular.
(*3) The complement of a regular languge is regular.
(*4) The difference of two regular languages is regular.
(*5) The reversal of a regular language is regular.
(*6) The closure (star) of a regular language is regular.
(*7) The concatenation of a regular language is regular.
(*8) A homomorphism (substitution of strings for symbols) of a regular language is regular.
(*9). The inverse homomorphism of a regular language is regular.

(*1). If L and M are regular languages, then so is $L \cup M$.

Proof:

L and M are regular languages ∴ they have reg. expressions

$\Rightarrow L = L(R)$; $M = L(S)$.

Then $L \cup M = L(R+S)$ by the def. of the + operator for regular expression.

(*2). If L is a regular language over alphabet Σ, then $\bar{L} = \Sigma^* - L$ is also a regular language.

Proof:

Let $L = L(A)$ for some DFA $A = (Q, \Sigma, \delta, q_0, F)$.

$\bar{L} = L(B)$ for $B = (Q, \Sigma, \delta, q_0, Q-F)$.

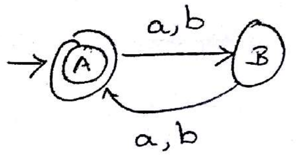
B is exactly like A, but the accepting states of A have become non-accepting states of B & vice versa.

$\Rightarrow w$ is in $L(B)$ if & only if $\hat{\delta}(q_0, w)$ is in $Q-F$, which occurs if & only if w is not in $L(A)$.

Scanned by CamScanner

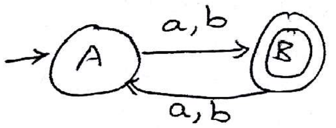
Ex:

①. $L_1 = \{$ strings of even length $\}$

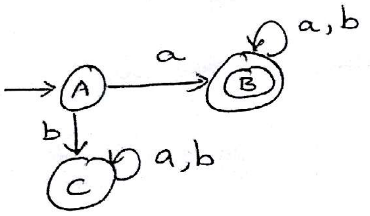


$L_2 = \overline{L_1}$

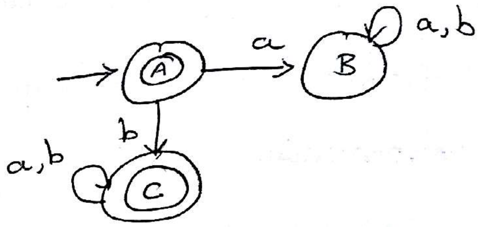
$= \{$strings of odd length$\}$



②. $L_3 = \{$strings starting with 'a'$\}$



$L_4 = \overline{L_3}$



③. If L and M are regular languages, then so is $L \cap M$.

Proof:

L and M are regular languages (given).

$A_L = (Q_L, \Sigma, \delta_L, q_L, F_L)$

$A_M = (Q_M, \Sigma, \delta_M, q_M, F_M)$.

$\Sigma \Rightarrow$ assume it to be same

if it is diff then take union.

It works for both NFA and DFA.

$$A_{L \cap M} = (Q_L \times Q_M ; \Sigma, \delta, (q_L, q_M), F_L \times F_M)$$

where $\delta((p,q),a) = (\delta_L(p,a), \delta_M(q,a))$

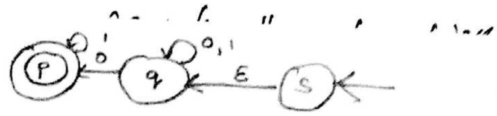
$$\therefore L(A_{L \cap M}) = L(A_L) \cap L(A_M)$$

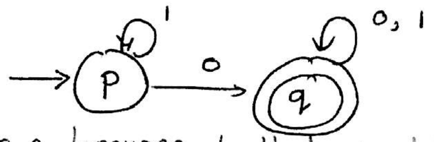
$$\hat{\delta}((q_L, q_M), \omega) = (\hat{\delta}(q_L, \omega), \hat{\delta}_M(q_M, \omega))$$

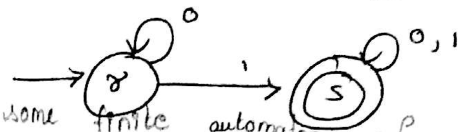
ω is accepted by A if & only if both A_L and A_m accept ω.

Thus A accepts the intersection of L and M.

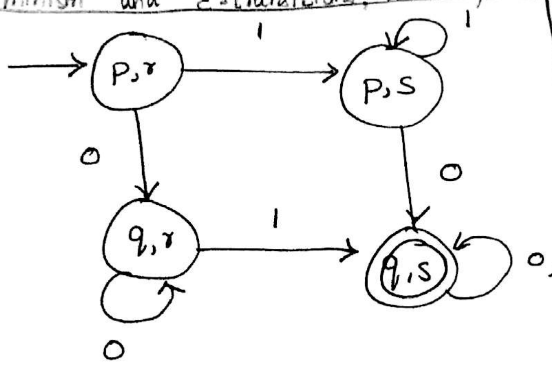
Ex: $L = 1^* 0 (0+1)^*$ $\quad L^R = (0+1)^* 0 1^*$







Given a language L that is $L(A)$ for some finite automaton, perhaps with non-determinism and ϵ-transitions, we may construct an automaton for L^R by,



1. Reverse all arcs in the transition diagram for A.
2. Make the start state of A be the only accepting state for the new automaton.
3. Create a new start state p_0 with transitions on ϵ to all the accepting states of A.

(*4). If L and M are regular languages, then so is $L-M$.

Proof:

$$L - M = L \cap \bar{M}$$

$\bar{M}$ is regular and $L \cap \bar{M}$ is also regular

so $L-M$ is also regular.

(*5). If L is a regular language, so is L^R.

Proof:

$E \Rightarrow$ Regular Exp.

$E \Rightarrow \epsilon \qquad E^R = \epsilon$

$\phi \qquad \phi$

$a \qquad a$

$\therefore E = E^R$.

$E = E_1 + E_2$ then $E^R = E_1^R + E_2^R$.

$E = E_1 . E_2$ then $E^R = E_2^R . E_1^R$.

Using Kleeni theorem

$L(E_1) = \{01, 111\}$ $\qquad L(E_2) = \{00, 10\}$

$L(E_1 E_2) = \{0100, 0110, 11100, 11110\}$

$(L(E_1 E_2))^R = \{0010, 0110, 00111, 01111\}$

$L(E_2)^R . L(E_1)^R = \{00, 01\}\{10, 111\} = \{0010, 00111, 0110, 01111\}$

$E = E_1^*$ then $E^R = (E_1^R)^*$.

w in $L(E)$

$$w_i = w_1 w_2 \dots w_n$$

$$w^R = w_n^R \, w_{n-1}^R \dots w_1^R .$$

Each w_i^R is in $L(E^R)$ so w^R is in $(E_1^R)^*$.

Any string in $L(E_1^R)^*$ is of the form $w_1 w_2 \dots w_n$, where each w_i is the reversal of a string in $L(E_1)$.

Reversal of the string $w_n^R \, w_{n-1}^R \dots w_1^R$ is $\therefore$ a string in $L(E_1^*)$ which is $L(E)$.

Thus string is in $L(E)$ if & only if its reversal is in $L(E_1^R)^*$.

Ex: R.E $L = (0+1)0^*$

$$L = 0^*(0+1).$$

(*8). If L is a regular language over alphabet Σ, and h is a homomorphism on Σ, then $h(L)$ is also regular.

Proof:

Let Σ and Γ are set of alphabets.

$$h : \underset{\substack{\downarrow \\ \text{single alph}}}{\Sigma} \rightarrow \underset{\substack{\downarrow \\ \text{string}}}{\Gamma^*}$$

If $w = a_1 a_2 a_3 \dots a_n$

$$h(w) = h(a_1)\, h(a_2) \dots h(a_n).$$

If L is made of alphabets from Σ, then $h(L) = \{h(w) \mid w \in L\}$ is called homomorphic image.

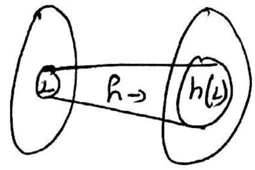
Ex:

① . $\Sigma = \{0, 1\}$ $\qquad h(0) = 01$ → } string

$\Gamma = \{0, 1, 2\}$ $\qquad h(1) = 112$ →

alphabets

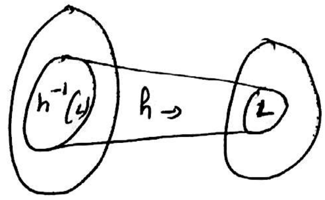
What is $h(010) = ?$.



$$h(010) = h(0)\,h(1)\,h(0)$$
$$= 01\ 112\ 01$$

② . $\Sigma = \{0, 1\}$ $\qquad h(0) = 3122$

$\Gamma = \{1, 2, 3\}$ $\qquad h(1) = 132$



What is $h(010) = ?$.

$$h(010) = 3122\ 132\ 3122.$$

(*9) . If h is a homomorphism from alphabet Σ to alphablet Γ and L is a regular language over Γ, then $h^{-1}(L)$ is also a regular language.

[If L is regular & h is homomorphism, then homomorphic image $h(L)$ is regular]

Proof :

Let R be reg. exp & $L(R)$ is reg. lang.

$h(R)$ can be found using $h(a)$ for each a in Σ.

By. def. $h(R)$ is a reg. exp. & so $h(L)$ is reg. lang.

∴ Reg. lang. is closed under homomorphism.

Ex : If $L = \{00, 010\}$ what is homormorphic

① image of L ?

$$L(00, 010) = L(h(00), h(010))$$
$$= L(h(0)h(0), h(0)h(1)h(0))$$
$$= (0101, 0111201)$$

$h(0) = 3122$ $\quad h(1) = 132$

③ $(0+1)^*(00)^* = (h(0) + h(1))^*\,(h(0)\,h(0))^*$

$$= (3122 + 132)^*\,(3122\,3122)^*$$

Ex: Suppose h is the homomorphism from the alphabet $\{0,1,2\}$ to the alphabet $\{a,b\}$ defined by:

$h(0) = a$; $h(1) = ab$; $h(2) = ba$.

(a). What is $h(0120)$? aabbaa

(b). What is $h(21120)$? baababbaa

(c). If L is the lang. $L(01^*2)$ what is $h(L)$? $a(ab)^*ba$.

(d). $L(0+12)$ " $(a + abba)$

(e). Suppose L is the language $\{ababa\}$, that is, the language consisting only the one string ababa. What is $h^{-1}(L)$?

$\underset{1}{\underline{ab}}\,\underset{1}{\underline{ab}}\,\underset{0}{\underline{a}}$ $\underset{0}{\underline{a}}\,\underset{2}{\underline{ba}}\,\underset{2}{\underline{ba}}$ $\underset{1}{\underline{ab}}\,\underset{0}{\underline{a}}\,\underset{2}{\underline{ba}}$

$h^{-1}(L) = \{110, 022, 102\}$.

Chomsky Hierarchy:

①. Type 3 : (Regular Grammar)

$A \rightarrow \alpha B \mid \beta$ Light Linear Grammar $A \rightarrow B\alpha \mid \beta$

$A, B \in V$ $\Leftarrow$ Right Linear Grammar $A, B \in V$

$\alpha, \beta \in T^*$ $\alpha, \beta \in T^*$

②. Type 2 : (CFG)

$A \rightarrow \alpha$, $A \in V$

$\alpha \in (V \cup T)^*$

③. Type 3 : (CSG)

$\alpha \rightarrow \beta$, $|\beta| \geq |\alpha|$