Context free Grammar (CFG)

- Can recognize both regular and non regular languages.
- Represented using 4 tuples, $G = (V, T, P, S)$ where,

  V : set of variables (in capital)

  T : set of terminals (in small)

  P : set of production $A \to \alpha$; $A \in V$ and $\alpha \in (V \cup T)^*$

  S : Start symbol

- Languages accepted by CFGs are called CFLs.

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

ϵ, ab, aabb, aaabbb, ...

$S \to aSb \mid \epsilon$

recursive substitution

For aabb, $S \to aSb$

$= aaSbb = aa\epsilon bb = aabb$

here,

$V = \{S\}$ $\quad P = \{ S \to aSb \mid \epsilon \}$

$T = \{a, b\}$ $\quad S = S$

If in this ques^n, instead of $n \geq 0$, $n \geq 1$ is given, then, replace 'ϵ' in the answer by 'ab'

If instead of $a^n b^n$, we have $a^n b^{2n}$, then, $S \to aSbb \mid \epsilon$

$L = \{a^n b^m \; ; \; n, m \geq 0\} \to$ regular

If $n = m \Rightarrow$ non regular

Here,

$$P = \left\{ \begin{array}{l} S \to AB \\ A \to aA \mid \epsilon \\ B \to bB \mid \epsilon \end{array} \right\}$$

and, $V = \{S, A, B\}$
$T = \{a, b\}$
$S = S$

(2) $L = \{a^n b^{n-3} \; ; \; n \geq 3\}$

aaa, aaaab, aaaaabb, ...

$S \to aSb \mid aaa$

(3) $L = \{a^n b^m \; ; \; n \neq m\}$

$S \to aSb \mid A \mid B$
$A \to aA \mid a$
$B \to bB \mid b$

(4) $L = \{a^n b^{n+m} c^m \; ; \; n, m \geq 0\}$

$L = a^n b^n b^m c^m$

$S \to S_1 S_2$
$S_1 \to aS_1b \mid \epsilon$
$S_2 \to bS_2c \mid \epsilon$

(5) $L = \{a^{n+m} b^n c^m \; ; \; n, m \geq 0\}$

$L = \underbrace{a^m \underbrace{a^n b^n} c^m}$

$S \to S_1$
$S_1 \to aS_1c \mid \epsilon \mid S_2$
$S_2 \to aS_2b \mid \epsilon$

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

$S \to aSa \mid bSb \mid \epsilon$

abba

Q. Write CFG for following languages:

(1) $L(G) = \{ ab(bbaa)^n bba(ba)^n \; ; \; n \geq 0 \}$

(2) $L = \{ a^n w w^R b^n \; ; \; n \geq 0, \; w \in \{0,1\}^* \}$

(3) To generate 0 or more as. (ie. a^*)

(4) Strings of palindromes

(5) To generate integers

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

(7) $L = \{ a^{n+2} b^m \; ; \; n \geq 0, \; m \geq n \}$

(8) $L = \{ a^n b^m c^K \; ; \; n + 2m = K \}$

(9) $L = \{ a^n b^m c^K \; ; \; n = m \text{ or } m \leq K \}$

(10) $L = \{ a^n b^m c^K \; ; \; n = m \text{ or } m \neq K \}$

(11) $L = \{ a^n b^m c^K \; ; \; K = |n - m| \}$

(12) $w a^n b^n w^R$

① ~

$S \to abS_1$

$S_1 \to bbaa S_1 ba \mid bba$

② ~

$S \to aSb \mid S_1 \mid \epsilon$

$S_1 \to aS_1a \mid bS_1b \mid \epsilon$

③ ~

$S \to aS \mid \epsilon$

④ ~ $S \to aSa \mid bSb \mid a \mid b \mid \epsilon$

⑤ ~ Let S_1 be the start symbol.

$S_1 \to SN$

$S \to + \mid - \mid \epsilon$

$N \to ND \mid D$

$D \to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

⑥ ~ $S \to aSb \mid bSa \mid \epsilon \mid SS$

⑦ ~ $S \to aaS_1 \cancel{B}$ IMP

$S_1 \to aS_1b \mid \epsilon \mid \cancel{B} S_2$

$S_2 \to b\cancel{B} \mid \epsilon$

$S \to aSa \mid bSb \mid S_1 \mid \epsilon$

$S_1 \to aS_1b \mid \epsilon$

⑪ ~ $\pm K = n - m$ $\begin{cases} n = m + K \\ m = n + K \end{cases}$

$a^K a^m b^m c^K$ or $a^n b^n b^K c^K$

$S \to S_1 \mid S_2$

$S_1 \to aS_1c \mid S_3 \mid \epsilon$

$S_3 \to aS_3b \mid \epsilon$

$S_2 \to S_3S_4$

$S_4 \to bS_4c \mid \epsilon$

for $a^n c b^n$

$S \to aSb \mid c$

⑧ ~ $L = a^n b^m c^K$

$= a^n b^m c^{n+2m}$

$= a^n b^m c^{2m} c^n$

$S \to aSc \mid S_2 \mid \epsilon$

$S_2 \to bS_2c^2 \mid \epsilon$

⑨ ~

$S \to S_1C \mid AS_2$

$S_1 \to aS_1b \mid \epsilon$

$S_2 \to bS_2c \mid C$

$C \to cC \mid \epsilon$

$A \to aA \mid \epsilon$

⑩ ~ $S \to S_1C \mid \cancel{BS_2} AS_2$

$S_1 \to aS_1b \mid \epsilon$

$C \to Cc \mid c$

$S_2 \to bS_2c \mid C \mid B$

$B \to bB \mid b$

$A \to aA \mid \epsilon$

Left most derivation & right most derivation

How we derive a string from a production is known as derivation.

LMD → The left most variable is derived first

RMD → The right most variable is derived first

Eg: If

$$\left.\begin{array}{l} E \to E+E \\ E \to E*E \\ E \to id \end{array}\right\} \text{productions}$$

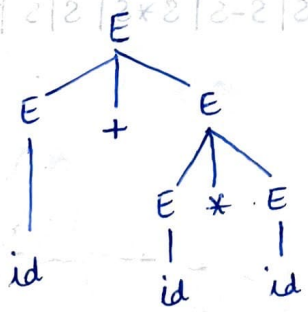
then derive id + id * id.

LMD	RMD
$E \to E+E$	$E \to E+E$
$E \to id+E$	$E \to E+E*E$
$E \to id+E*E$	$E \to E+E*id$
$E \to id+id*E$	$E \to E+id*id$
$E \to id+id*id$	$E \to id+id*id$
(left to right)	(right to left)

Intermediate steps are called sentential form (excluding the start symbol and the derived string)

Derivation tree

For RMD



Parse tree

The root node of a derivation tree is the start symbol.

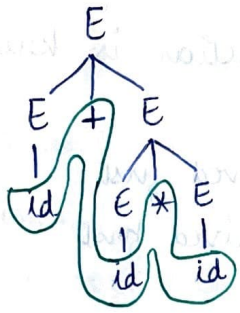
The leaf nodes of derivation tree are terminus.

Yield of a tree

All leaf nodes of the tree from left most corner of the tree to right (for both LMD and RMD) give the string that we have derived.

Eg: id + id * id

LMD :



Yield : id + id * id

If we can obtain 2 LMDs or 2 RMDs for the same string (from the same productions), then the grammar is said to be ambiguous grammar.

Eg :

LMD_2

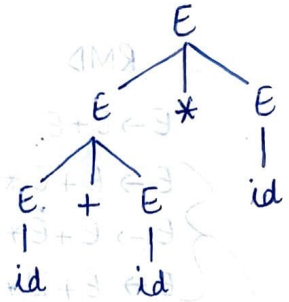
$E \to E*E$

$E \to E+E*E$

$E \to id+E*E$

$E \to id+id*E$

$E \to id+id*id$



∴ the grammar is ambiguous.

Q. Prove that the following grammars are ambiguous :

(1) ~~a~~ $S \to aS \mid X$

$X \to aX \mid a$

(2) $S \to aSbS \mid bSaS \mid \epsilon$

(3) $S \to AB \mid aaB$

$A \to a \mid Aa$

$B \to b$

(4) $S \to SS \mid aSb \mid bSa \mid \epsilon$

(5) $S \to S+S \mid S-S \mid S*S \mid S/S \mid (S) \mid a$

(1) LMD_1

$S \to aS$

$\to aaS$ (as $S \to aS$)

$\to aaX$ (as $S \to X$)

$\to aaa$ (as $X \to a$)

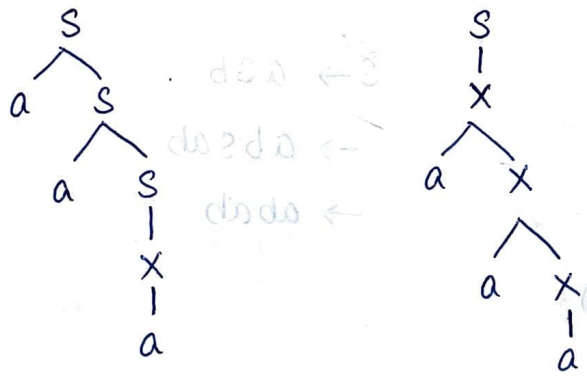
LMD_2

$S \to X$

$\to aX$ (as $X \to aX$)

$\to aaX$ (as, $X \to aX$)

$\to aaa$ (as, $X \to a$)



As derivation trees are not same, the grammar is ambiguous

②

$S \to aSbS$

$S \to abS \quad (S \to \epsilon)$

$S \to abasbS \quad (S \to asbS)$

$S \to ababS \quad (S \to \epsilon)$

$S \to ababbSaS \quad (S \to bSaS)$

$S \to ababbaS \quad (S \to \epsilon)$

$S \to ababba \quad (S \to \epsilon)$

~~$S \to bSaS \quad bSb$~~

$S \to aSbS$

$S \to absasbS \quad (S \to bSaS)$

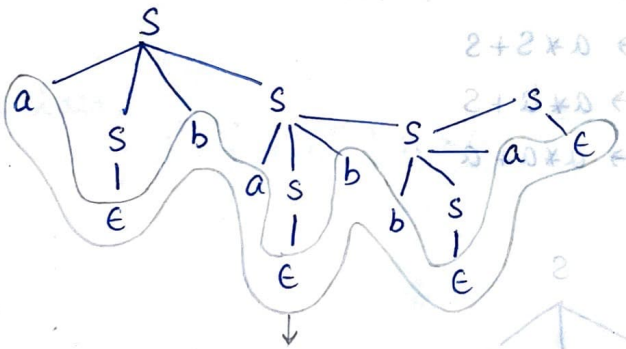
$S \to abasbS \quad (S \to \epsilon)$

$S \to ababS \quad (S \to \epsilon)$

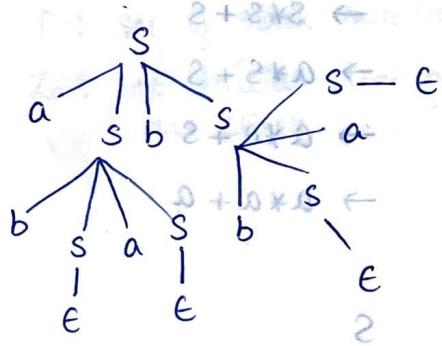
$S \to ababbSaS \quad (S \to bSaS)$

$S \to ababbaS \quad (S \to \epsilon)$

$S \to$ ~~to~~ $ababba \quad (S \to \epsilon)$



Yield of the tree



③

$S \to AB$

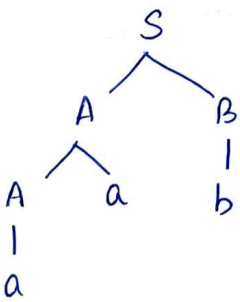
$\to AaB \quad (A \to Aa)$

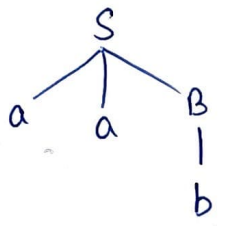
$\to aaB \quad (A \to a)$

$\to aab \quad (B \to b)$

$S \to aaB$

$\to aab \quad (B \to b)$





④~

$S \to SS$

$\to S$

$\to aSb$

$\to absab$

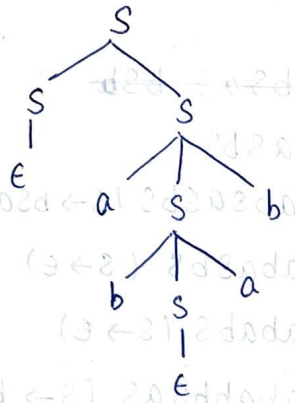
$\to abab$

$\to aSbS \quad (S \to aSb)$

$\to abS \quad (S \to \epsilon)$

$\to abbSa \quad (S \to bSa)$

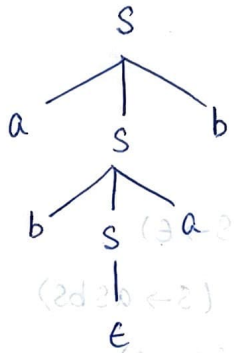
$\to abba \quad (S \to \epsilon)$



$S \to aSb$

$\to absab$

$\to abab$



⑤~

$S \to S+S$

$\to S*S+S$

$\to a*S+S$

$\to a*a+S$

$\to a*a+a$

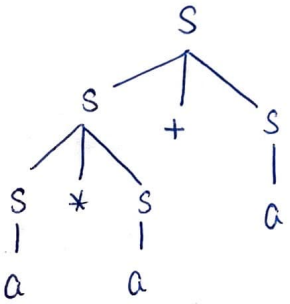
$S \to S*S$

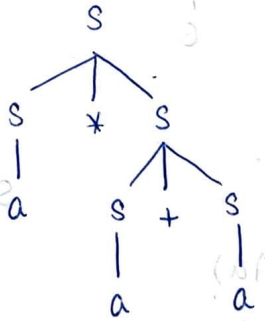
$\to a*S$

$\to a*S+S$

$\to a*a+S$

$\to a*a+a$





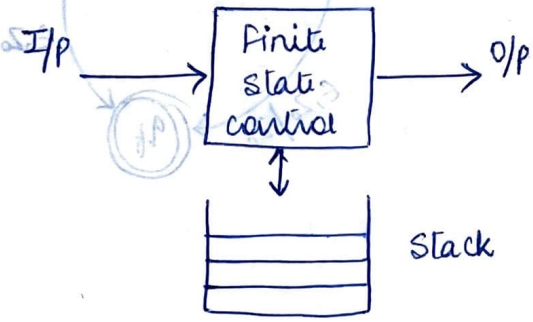
Simplification of CFG

- Chomsky normal form (CNF)
- Greibach normal form (GNF)

* Push down automata (PDA)

- Defined using 7 tuples.

$$M = (Q, \Sigma, \delta, q_0, F, \Gamma, Z_0)$$



Q : set of states

Σ : set of input alphabet

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

q_0 : Initial state

F : set of final states

Γ : set of stack symbols

Z_0 : ~~set~~ Initial element on the stack. TOS

Consider,

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

Eg: a a b b ϵ

Transitions

$\delta(q_0, \epsilon, Z_0) = (q_f, Z_0) \rightarrow$ for $n=0$

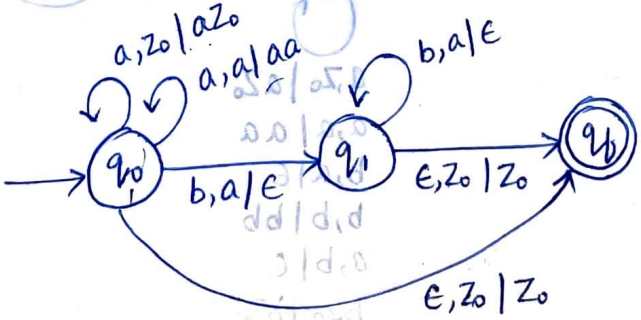
$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

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

$\delta(q_0, b, a) = (q_1, \epsilon)$ → represents pop operation

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

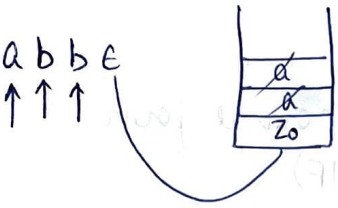
$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$



* $L = \{a^n b^{2n} \mid n \geq 0\}$

For every 'a', push 2 a's and pop 1 'a' for every 'b'.

a b b ε



It makes the no. of as and bs equal.

$\delta(q_0, \epsilon, Z_0) = (q_f, Z_0)$

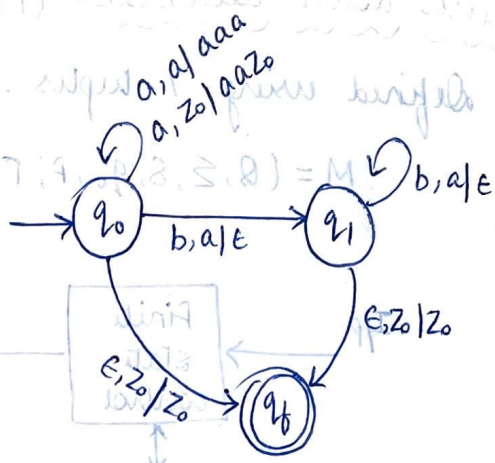
$\delta(q_0, a, Z_0) = (q_0, aaZ_0)$

$\delta(q_0, a, a) = (q_0, \underbrace{aaa}_{\text{push}})$

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

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

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



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

Input	Top of stack	Operation
a	Z_0	Push
a	a	Push
a	b	Pop
b	Z_0	Push
b	b	Push
b	a	Pop

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

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

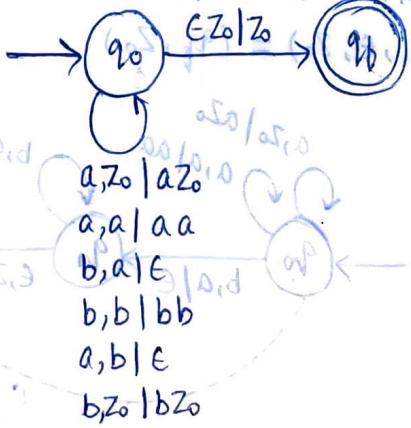
$\delta(q_0, b, a) = (q_0, \epsilon)$

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

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

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

$\delta(q_0, \epsilon, Z_0) = (q_f, Z_0)$



Q. Write PDA for the following languages :

(1) $L = \{ a^n b^{n+m} c^m \mid n \geq 0,\ m \geq 1 \}$

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

(3) $L = \{ n_a(w) = n_b(w) + 1 \mid w \in \{a,b\}^* \}$

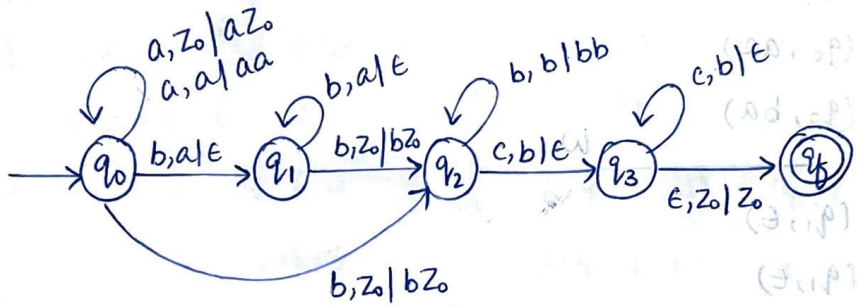
(4) $L = \{ w c w^R \mid w \in (a,b)^* \}$ // palindrome

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

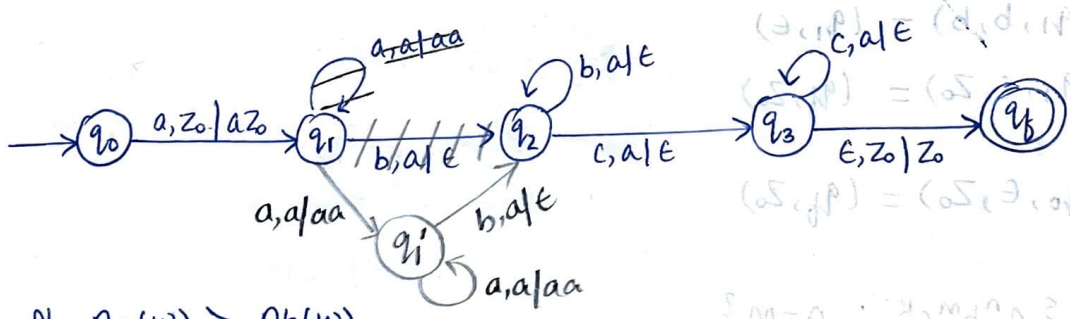
(6) $L = \{ a^n b^m c^K \mid n = m \}$

(7) $L = \{ a^n b^m c^K \mid K = n+m \}$

①

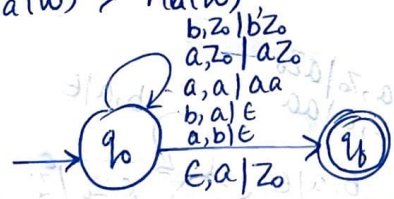


②

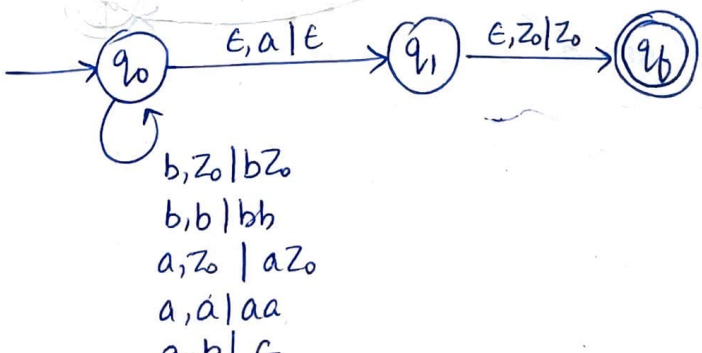


③

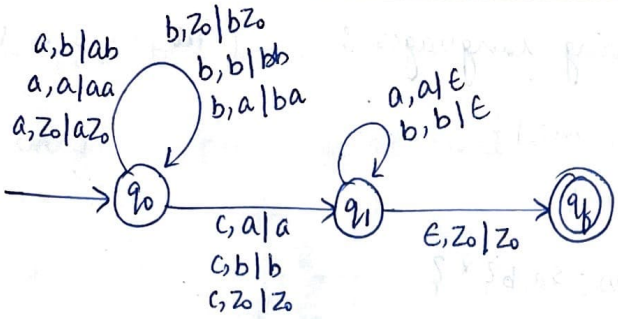
If $n_a(w) > n_b(w)$



when, $n_a(w) = n_b(w) + 1$



④



⑤ Non deterministic

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

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

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

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

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

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

Eg: $\overset{w}{aaba} \mid \overset{w^R}{abaa}$

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

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

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

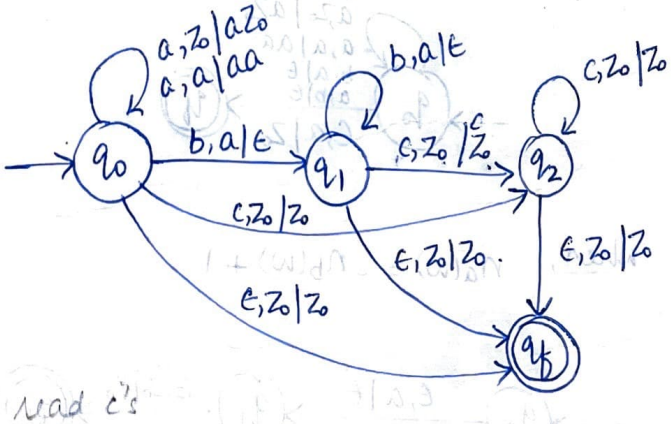
$\delta(q_1, b, b) = (q_1, \epsilon)$

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

$\delta(q_0, \epsilon, Z_0) = (q_f, Z_0)$

⑥ $L = \{a^n b^m c^k \; ; \; n = m\}$

or, $L = a^n b^n c^k$



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

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

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

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

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

$\delta(q_2, c, Z_0) = (q_2, Z_0)$

Just read c's
Do not push them on the stack

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

$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$ → no cs

$\delta(q_0, \epsilon, Z_0) = (q_f, Z_0)$ → no as & bs or cs (ϵ)

$\delta(q_0, c, Z_0) = (q_2, Z_0)$ // only cs.

Instantaneous description (ID)

ID is represented by triplet (q, w, u) where,

q : current state
w : unscanned input
u : contents of stack

A move from one ID to another is denoted by $\vdash$

$(q, aw, bx) \vdash (q_2, w, yx)$ is possible if and only if $\delta(q, a, b) = (q_2, y)$

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

$\delta(q_0, \epsilon, Z_0) = (q_f, Z_0)$
$\delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, b, Z_0) = (q_0, bZ_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_0, ba)$
$\delta(q_0, a, b) = (q_0, ab)$

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

$(q_0, \underbrace{abb}_{w}\underbrace{bba}_{w^R}, Z_0) \vdash (q_0, bbbba, aZ_0)$

$\vdash (q_0, bbbba\not\epsilon, baZ_0)$

push: $\vdash (q_0, bba, bbaZ_0)$

pop: $\vdash (q_1, bba, aZ_0)$ X — such a comb.^{n} is not possible!

push: $\vdash (q_0, ba, bbbaZ_0)$ X — not enough chars. in the unscanned i/p to pop stack contents

pop: $\vdash (q_1, ba, baZ_0)$
$\vdash (q_1, a, aZ_0)$
$\vdash (q_1, \epsilon, Z_0)$
(q_f, Z_0)

For <u>aabb</u> :

$(q_0, aabb, Z_0) \vdash (q_0, abb, aZ_0)$
$\vdash (q_0, bb, aaZ_0) \vdash (q_0, b, baaZ_0) \vdash (q_0, \epsilon, bbaaZ_0)$

Languages of PDA

(1) Acceptance by final state :

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

Let $P = \{Q, \Sigma, \delta, q_0, F, \Gamma, Z_0\}$, then $L(P)$: language accepted by P by final state is given by,

$\{ w \mid (q, w, Z_0) \vdash_P^* (q, \epsilon, \alpha) \}$ for some state q in F and any stack string 'α' starting with initial ID w waiting on i/p. P consumes w from input and enters the accepting state.

Contents of stack at that time are irrelevant.

(2) Acceptance by empty stack :

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

" same defn " but, $\{ w \mid (q, w, Z_0) \vdash_P^* (q, \epsilon, \epsilon) \}$

Q. Write a PDA for acceptance by empty stack for the language

$L = \{ n_a + n_b = n_c \}$

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

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

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

$\delta(q_0, c, Z_0) = (q_0, cZ_0)$

$\left.\begin{matrix} ac \\ ca \\ bc \\ cb \end{matrix}\right\} \rightarrow$ pop

$\left.\begin{matrix} ab \\ ba \\ cc \\ aa \\ bb \end{matrix}\right\} \rightarrow$ push

* Chomsky normal form : (To simplify CFG)

Consider the example,

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

and,

$S \rightarrow aSb \mid \epsilon$

$A \rightarrow aA \mid B \mid \epsilon$

$B \rightarrow bB \rightarrow$ cannot derive a terminal, hence useless

useless symbols ~~useless states~~ $\downarrow$ cannot be reached from the start state

Algorithm

* Find the set of symbols from which the string of ~~to~~ terminals can be obtained using following algo :

Step(1)

$OV = \phi$

$NV = \{ A \mid A \rightarrow w \mid w \in T^* \}$

while $(OV \neq NV)$

$OV = NV$;

$NV = OV \cup \{ A \mid A \rightarrow \alpha, \alpha \in (OV \cup T)^* \}$

end while

$V_1 = NV$

All the productions used to get V_1 are in production P_1.

Step 2) Obtain symbols which are reachable from the start symbol.

$V' = \{S\}$

```
for each A in V'
    if A → α
        add var to V1'
        add Ter to T'
    endif
endfor
```

Q. (1) $S \to aAB$

$A \to aB \mid B$

$B \to aB \mid b \mid bC$

$D \to Ea$

$E \to a \mid aE \mid bC$

can't be reached from S, hence useless symbols.

here, 'a' and 'b' are terminals.

Step (1)

OV	nV	P_1
ϕ	B, E	$B \to b$, $E \to a$
B, E	A, B, D, E	$A \to aB \mid B$, $B \to aB$, $D \to Ea$, $E \to aE$
A, B, D, E	A, B, D, E, S	$S \to aAB$
A, B, D, E, S	A, B, D, E, S	–

same, hence stop

Step (2)

$V' = \{S\}$

	T'	V'
$S \to aAB$	a	A, B
$A \to aB \mid B$	a	A, B
$B \to aB \mid b$	a, b	A, B

Ans.

(2) $S \to AB \mid aAC$
$A \to aA \mid b$
$B \to bB \mid a$
$D \to aA \mid bB$
$E \to aD \mid bD$

Terminals
↓
Symbols that only derive alphabets and ϵ

OV	nv	P_1
ϕ	A, B	$A \to b$, $B \to a$
A, B	A, B, S, D	$A \to aA$, $B \to bB$, $D \to aA \mid bB$, $S \to AB$
A, B, S, D	A, B, S, D, E	$E \to aD \mid bD$
A, B, S, D, E	A, B, S, D, E	–

	T'	V'
$S \to AB$	–	A, B
$A \to b \mid aA$	a, b	–
$B \to a \mid bB$	a, b	–

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

(3) $S \to AB \mid CA$
$A \to a$
$B \to BC \mid AB$
$C \to aB \mid b$

(4) $S \to A$
$A \to aA \mid \epsilon$
$B \to bA$

(5) $S \to a \mid aA \mid B \mid C$
$A \to aB \mid \epsilon$
$B \to Aa$
$C \to cCD$
$D \to ddd$

(3)

OV	nv	P_1
ϕ	A, C	$A \to a$, $C \to b$
A, C	A, C, S	$S \to CA$
A, C, S	A, C, S	–

	T'	V'
$S \to CA$	–	A, C
$A \to a$	a	–
$C \to b$	b	–

④

OV	nv	P_1
ϕ	A	$A \to \epsilon$
A	A, S, B	$S \to A$, $A \to aA$, $B \to bA$
A, B, S	A, B, S	–

	T'	V'
$S \to A$	–	A
$A \to aA \mid \epsilon$	a	A

⑤

OV	nv	P_1
ϕ	S, A, D	$S \to a$, $A \to \epsilon$, $D \to ddd$
S, A, D	S, A, B, D	$S \to aA$, $B \to Aa$
S, A, B, D	S, A, B, D	$A \to aB$, $S \to B$

	T'	V'
$S \to a \mid aA \mid B$	a	A, B
$A \to aB \mid \epsilon$	a	A, B
$B \to Aa$	a	A, B

27/03/13

Eliminating ϵ- production

$L(G) - \epsilon$ as ϵ is not accepted by any normal form.

Any production leading to ϵ is called nullable.

Eg:

$S \to A$

$A \to aA \mid \epsilon$

$B \to bA$

here, A is nullable variable and as S derives A, $\therefore$ S is also nullable.

Algorithm

Step 1: Initialize P_1 to P.

Step 2: Find all nullable variables in P.

Step 3: For every production $A \to \alpha$ in P, add to P_1 every production that can be obtained from it by deleting from α one or more occurences of nullable variable.

Step 4: Delete all ϵ- productions from P_1 and delete all duplicate productions and productions of the form $A \to A$

Algo. to find nullable variables :

$N_0 = \{A \in V \mid P \text{ productions } A \to \epsilon\}$

$i = 0$;

do

$i = i+1$;

$N_i = N_{i-1} \cup \{A \mid A \to \alpha, \alpha \in N_{i-1}^*\}$

while $N_i \neq N_{i-1}$

N_i is set of nullable variables.

Q. (1) $S \to AB$

$A \to aAA \mid \epsilon$

$B \to bBB \mid \epsilon$

here, $N_0 = \{A, B\}$

$N_1 = N_0 \cup S = \{A, B, S\}$

↳ nullable variables

Step 1 ~

$$P_1 = \begin{cases} S \rightarrow \\ A \rightarrow \\ B \rightarrow \end{cases}$$

Step 2 ~ Nullable : A, B, S

Step 3 ~

$$P_1 = \begin{cases} S \rightarrow AB \mid B \mid A \mid \cancel{\epsilon} \\ A \rightarrow aAA \mid \cancel{\epsilon} \mid aA \mid \cancel{aA} \mid a \\ B \rightarrow bBB \mid \cancel{\epsilon} \mid bB \mid \cancel{bB} \mid b \end{cases}$$

Step 4 ~

$$P_1 = \begin{cases} S \rightarrow AB \mid B \mid A \\ A \rightarrow aAA \mid aA \mid a \\ B \rightarrow bBB \mid bB \mid b \end{cases}$$

(2)

$S \rightarrow ABBAD$

$A \rightarrow aB \mid \epsilon$

$B \rightarrow bA \mid \epsilon$

$D \rightarrow a$

$N = \{A, B\}$

$S \rightarrow ABBAD \mid BBAD \mid ABBD \mid ABAD \mid \cancel{ABAD} \mid BAD \mid \cancel{AAD} \mid \cancel{ABD} \mid BBD \mid AD \mid BD \mid D$

$A \rightarrow aB \mid \cancel{\epsilon} \mid a$

$B \rightarrow bA \mid \cancel{\epsilon} \mid b$

$D \rightarrow a$

(3)

$S \rightarrow aAa \mid AB$

$A \rightarrow BS \mid aBa \mid \epsilon$

$B \rightarrow \epsilon \mid ab$

$N = \{A, B, S\}$

$S \rightarrow aAa \mid AB \mid aa \mid B \mid A \mid \cancel{\epsilon}$

$A \rightarrow BS \mid aBa \mid \cancel{\epsilon} \mid S \mid aa \mid B$

$B \rightarrow \cancel{\epsilon} \mid ab$

Elimination of unit productions

A production is said to be a unit production if it is of the type:

$S \to T$; $S, T \in V$ ↳ variables

Algorithm

Step (1) : Initialize P_1 to P

Step (2) : For each $A \in V$, find the set of A derivables

Step (3) : For every pair (A, B), s.t., B is A derivable and every non unit production $B \to \alpha$, add the production $A \to \alpha$ to P_1 if it is not present in P_1

Step (4) : Delete all unit productions from P_1.

Questions

(1)

$$P_1 = \begin{cases} S \to S+T \mid T \\ T \to T*F \mid F \\ F \to (S) \mid a \end{cases}$$

here, $\Sigma = \{a, (,), *, +\}$

$S \to T$

$T \to F$

After eliminating unit productions,

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

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

$F \to (S) \mid a$

(2) $S \to AB$

$A \to a$

$B \to C \mid b$

$C \to D$

$D \to E \mid bc$

$E \to d \mid Ab$

$$\left.\begin{array}{l} B \to C \\ C \to D \\ D \to E \end{array}\right\} B \to E$$

$\therefore D \to d \mid Ab \mid bc$

If $A \to B$, Then B is A derivable.

$S \to AB$

$A \to a$

$B \to d \mid Ab \mid \cancel{bc}\ bc \mid b$

$C \to bc \mid d \mid Ab$

$D \to d \mid Ab \mid bc$

$E \to d \mid Ab$

Chomsky normal form (CNF)

A production is said to be in CNF if it is of the form:

$A \to BC$

$A \to a$

where, $A, B, C \in V$

and $a \in T$

V : variable

T : terminal

Why to convert in NF?

Easy to implement.

Q. Convert the following grammar in CNF:

(1) $S \to AACD$

$A \to aAb \mid \epsilon \mid ab$

$C \to aC \mid a$

$D \to aDa \mid bDb \mid \epsilon$

$S \to S+T$

$S \to SS_1$
$S_1 \to t_1 S_1$
$t_1 \to t$
} CNF

(1) No useless productions.

(2) A and D are nullable.

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

↳ unit prod^n

∴ write $a \mid a$ instead.

$A \to aAb \mid \cancel{\epsilon} \mid ab$

$C \to ac \mid a$

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

(3)

$S \to A_1A_2 \mid AA_2 \mid A_1C \mid CD \mid AC \mid A_3C \mid a$

$A_1 \to AA$

$A_2 \to CD$

$A_3 \to a$

$A_4 \to A_3A$

$A_5 \to b$

$A \to A_4A_5 \mid A_3A_5$

$C \to A_3C \mid a$

$D \to A_6A_3 \mid A_3A_3 \mid A_7A_5 \mid A_5A_5$

$A_6 \to A_3D$

$A_7 \to A_5D$

So there are 4 steps involved:

(1) Remove useless productions/symbols

(2) Eliminate ϵ productions

(3) Eliminate unit productions

(4) Convert to CNF

(2) $S \to aSb \mid ab$

(3) $S \to ab \mid aS \mid aaS$

(4) $S \to ABb \mid a$
$A \to aaA \mid B$
$B \to bAb$

② $S \to S_1S_2 \mid S_3S_2$
$S_1 \to S_3S$
$S_2 \to b$
$S_3 \to a$

③ ~~$S \to ab \mid a$~~
$S \to S_1S_2 \mid S_3 \mid S_3S_3$
$S_1 \to a$
$S_2 \to b$
$S_3 \to aS$

④ A and B are useless productions
(do not reach a terminal)

$\therefore S \to a$

05/04/13

(1) $S \to ABaC$
$A \to BC \mid \epsilon$
$B \to b \mid \epsilon$
$C \to D \mid \epsilon$
$D \to d$

(2) $S \to a \mid aA \mid B \mid C$
$A \to aB \mid \epsilon$
$B \to Aa$
$C \to cCD$
$D \to ddd$

(3) $S \to AaB \mid aaB$
$A \to \epsilon$
$B \to bbA \mid \epsilon$

(4) $S \to aA \mid aBB$
$A \to aaA \mid \epsilon$
$B \to bB \mid bbC$
$C \to B$

(5) $S \to abAB$
$A \to bAB \mid \epsilon$
$B \to BAa \mid A \mid \epsilon$

(6) $S \to AaA \mid CA \mid BaB$
$A \to aaBa \mid CDA \mid aa \mid DC$
$B \to bB \mid bAB \mid bb \mid aS$
$C \to Ca \mid bC \mid \cancel{a}D$
$D \to bD \mid \epsilon$

① After removing useless productions,

$S \to ABaC$
$A \to BC$
$B \to b \mid \epsilon$
$C \to D \mid \epsilon$
$D \to d$

} no useless pd^(s)

A, B, C are nullable variables

$S \to ABaC \mid AaC \mid ABa \mid Aa \mid a \mid BaC \mid \underline{AC} \mid Ba$
$A \to BC \mid B \mid C \mid \cancel{\epsilon}$
$B \to b \mid \cancel{\epsilon}$
$C \to D \mid \cancel{\epsilon}$
$D \to d$

After eliminating unit productions :

$S \to ABaC \mid AaC \mid BaC \mid ABa \mid \cancel{BaC} \mid aC \mid Ba \mid Aa \mid a$
$A \to BC \mid b \mid d$
$B \to b$
$C \to d$
$D \to d$

CNF :

$S \to A_1A_2 \mid AA_2 \mid BA_2 \mid A_1A_3 \mid A_3C \mid BA_3 \mid AA_3 \mid a$
$A_1 \to AB$
$A_2 \to A_3C$
$A_3 \to a$
$A \to BC \mid b \mid d$
$B \to b$

$C \to d$
$D \to d$

② After removing useless productions,

$S \to a \mid aA \mid B$
$A \to aB \mid \epsilon$
$B \to Aa$

A is nullable.

$S \to a \mid aA \mid B \mid \cancel{a}$
$A \to aB \mid \cancel{\epsilon}$

$B \to Aa \mid a$

After eliminating unit pd^n:

$S \to a \mid aA \mid Aa$

$A \to aB$

$B \to Aa \mid a$

CNF:

$S \to a \mid A_1A \mid AA_1$

$A \to A_1B$

$B \to AA_1 \mid a$

$A_1 \to a$

③

No useless productions

A and B are nullable variables

$S \to AaB \mid aaB \mid aB \mid Aa \mid a \mid aa$

$A \to \cancel{\epsilon}$

$B \to bbA \mid \cancel{\epsilon} \mid bb$

Now,

$S \to aaB \mid aB \mid a \mid aa$

$B \to bb$

No unit productions

CNF:

$S \to A_1A_2 \mid A_1B \mid a \mid A_1A_1$

$B \to B_1B_1$

$A_1 \to a$

$A_2 \to A_1B$

$B_1 \to b$

④

B and C are useless

$S \to aA$

$A \to aaA \mid \epsilon$

A is nullable

$S \to aA \mid a$

$A \to aaA \mid aa$

No unit productions.

$S \to A_1A \mid a$

$A_1 \to a$

$A \to A_1A_2 \mid A_1A_1$

$A_2 \to A_1A$

⑥~ No useless production

D, A, C, S → nullable variables

$S \to AaA \mid Aa \mid aA \mid a \mid CA \mid C \mid A \mid BaB \mid \cancel{\epsilon}$

$A \to aaBa \mid CDA \mid CD \mid DA \mid CA \mid C \mid D \mid A \mid aa \mid DC \mid \cancel{D} \mid \cancel{C}$

$B \to bB \mid bAB \mid \cancel{bB} \mid bb \mid as \mid a$

$C \to Ca \mid a \mid bC \mid b \mid D$

$D \to bD \mid b$

Eliminating unit productions :

$S \to AaA \mid Aa \mid aA \mid a \mid CA \mid \underbrace{Ca \mid a \mid bC \mid b}_{C} \mid \underbrace{bD \mid b}_{D} \mid \underbrace{aaBa \mid CDA \mid CD \mid DA \mid CA}_{A}$

$\underbrace{aa \mid DC}_{A} \mid BaB$

$A \to aaBa \mid CDA \mid CD \mid DA \mid CA \mid \underbrace{Ca \mid a \mid bC \mid b \mid bD}_{C} \mid aa \mid DC$

$B \to bB \mid bAB \mid bb \mid as \mid a$

$C \to Ca \mid a \mid bC \mid b \mid bD$

$D \to bD \mid b$

Converting to CNF :

$A_1 \to AA_2$ $\quad$ $A_5 \to A_2A_2$

$A_2 \to a$ $\quad$ $A_6 \to BA_2$

$A_3 \to CA$ $\quad$ $A_7 \to CD$

$A_4 \to b$ $\quad$ $A_8 \to AB$

$S \to A_1A \mid AA_2 \mid A_2A \mid a \mid CA \mid CA_2 \mid A_4C \mid b \mid A_4D \mid A_3A_6 \mid A_7A \mid CD \mid DA \mid CA \mid A_2A_2 \mid DC \mid A_6B \mid BA_2 \mid A_2B$

$B \to A_4B \mid A_4A_8 \mid A_4A_4 \mid A_2S \mid \cancel{A}\, a$

$C \to CA_2 \mid a \mid A_4C \mid b \mid A_4D$

$D \to A_4D \mid b$

$A \to A_5A_6 \mid A_7A \mid CD \mid DA \mid CA \mid CA_2 \mid a \mid A_4C \mid b \mid A_4D \mid A_2A_2 \mid DC$

Deterministic PDA

$PDA = (Q, \Sigma, \delta, q_0, \Gamma, Z_0, F)$

Eg: WCW^R

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

For DPDA, the following 2 conditions must be satisfied:

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

2) For any $q \in Q$, $x \in \Gamma$, if $\delta(q, \epsilon, x) \neq \phi$, then $\delta(q, a, x) = \phi$ for every $a \in \Sigma$

Eg: For WW^R:

$q_0 \; (a, a)$

$$\left.\begin{aligned} \delta(q_0, a, Z_0) &= (q_0, aZ_0) \\ \delta(q_0, b, Z_0) &= (q_0, bZ_0) \\ \delta(q_0, a, b) &= (q_0, ab) \\ \delta(q_0, b, a) &= (q_0, ba) \\ \delta(q_0, a, a) &= (q_0, aa) \\ \delta(q_0, b, b) &= (q_0, bb) \end{aligned}\right\} W$$

violates ① ∴ NPDA

$$\left.\begin{aligned} \delta(q_0, a, a) &= (q_1, \epsilon) \\ \delta(q_0, b, b) &= (q_1, \epsilon) \\ \delta(q_1, a, a) &= (q_1, \epsilon) \\ \delta(q_1, b, b) &= (q_1, \epsilon) \end{aligned}\right\} W^R$$

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

violates ② cond^n ∴ NPDA

$$\left.\begin{aligned} \delta(q_0, a, Z_0) &= (q_0, aZ_0) \\ \delta(q_0, a, a) &= (q_0, aa) \\ \delta(q_0, b, a) &= (q_1, \epsilon) \\ \delta(q_1, b, a) &= (q_1, \epsilon) \\ \delta(q_1, \epsilon, Z_0) &= (q_f, Z_0) \\ \delta(q_0, \epsilon, Z_0) &= (q_f, Z_0) \end{aligned}\right\} n \geq 0$$

Q. $L = \{n_a(w) = n_b(w)\}$

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

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

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

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

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

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

$\delta(q_0, \epsilon, Z_0) = (q_f, Z_0)$

violates cond^n (2)

∴ NPDA!

Equivalence of PDAs and CFGs

CFG → PDA

Algorithm

1) Let q_0 be the start state and Z_0 be the initial symbol in stack. In state q_0 without consuming any input, change the state to q_1 and push the start symbol S on the stack using the transition,

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

2) (i) For every $A \in V$, introduce the transition $\delta(q_1, \epsilon, A) = (q_1, \alpha)$ for all the productions of the form $A \to \alpha$ in G.

(ii) For every $A \in \Sigma$, introduce the transition, $\delta(q_1, a, a) = (q_1, \epsilon)$

3) Finally in state q_1, without consuming any i/p, change the state to q_f.

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

Eg: $S \to aSb \mid \epsilon$

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

2. $\delta(q_1, \epsilon, S) = (q_1, aSb)$ and $\delta(q_1, a, a) = (q_1, \epsilon)$

   $\delta(q_1, \epsilon, S) = (q_1, \epsilon)$ $\quad$ $\delta(q_1, b, b) = (q_1, \epsilon)$

   ~~$\delta(q_1, a, b) = (q_1, \epsilon)$~~

3. $\delta(q_1, \epsilon, Z_0) = (q_f, Z_0)$ $\quad$ ~~$\delta(q_1, b, a) = (q_1, \epsilon)$~~

①
$S \to aABB \mid aAA$
$A \to aBB \mid a$
$B \to bBB \mid A$
$C \to a$

②
$S \to aABC$
$A \to aB \mid b$
$B \to bA \mid a$
$C \to a$

①

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

2. (i) $\delta(q_1, \epsilon, S) = (q_1, aABB)$ $\quad$ $\delta(q_1, \epsilon, B) = (q_1, bBB)$

   $\delta(q_1, \epsilon, S) = (q_1, aAA)$ $\quad$ $\delta(q_1, \epsilon, B) = (q_1, A)$

   $\delta(q_1, \epsilon, A) = (q_1, aBB)$ $\quad$ $\delta(q_1, \epsilon, C) = (q_1, a)$

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

   (ii) $\delta(q_1, a, a) = (q_1, \epsilon)$ ~~$\delta(q_1, a, b) = (q_1, \epsilon)$~~ $\quad$ $\delta(q_1, b, b) = (q_1, \epsilon)$ ~~$\delta(q_1, b, a) = (q_1, \epsilon)$~~

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

PDA → CFG

Algorithm

1) Assume there is a transition from q_i to q_j on i/p symbol 'a' and top of stack 'Z'. The non terminal corresponding to this transition is given by, $\underbrace{(q_i Z q_j)}_{\text{single variable}}$

2) If there is a transition of the form $\delta(q_i, a, Z) = (q_j, AB)$, then introduce the production of the form : $(q_i Z q_K) \rightarrow a(q_j A q_l)(q_l B q_K)$, where q_l, q_K are all possible states of Q

3) If there is a transition of the form $\delta(q_i, a, Z) = (q_j, \epsilon)$, then introduce the production of the form $(q_i Z q_j) \rightarrow a$.

Ques.

① $\delta(q_0, a, Z) = (q_0, AZ)$

$\delta(q_0, b, A) = (q_0, AA)$ $\quad q_0$: initial

$\delta(q_0, a, A) = (q_1, \epsilon)$ $\quad q_1$: final

(i) Start symbol : $(q_0 Z q_1)$

(ii) (a) $(q_0 Z q_0) \rightarrow a(q_0 A q_0)(q_0 Z q_0) \mid a(q_0 A q_1)(q_1 Z q_0)$

For every q_K, vary q_l

$(q_0 Z q_1) \rightarrow a(q_0 A q_0)(q_0 Z q_1) \mid a(q_0 A q_1)(q_1 Z q_1)$

(b) $(q_0 A q_0) \rightarrow b(q_0 A q_0)(q_0 A q_0) \mid b(q_0 A q_1)(q_1 A q_0)$

$(q_0 A q_1) \rightarrow b(q_0 A q_0)(q_0 A q_1) \mid b(q_0 A q_1)(q_1 A q_1)$

(c) $(q_0 A q_1) \rightarrow a$

If $\delta(q_0, b, A) = (q_0, A)$ // not in standard form

then,

$\delta(q_0, \epsilon, A) = (q_2, AA) \rightarrow$ one push

$\delta(q_2, b, A) = (q_0, \epsilon) \rightarrow$ one pop

} does not change the language

1. Start symbol : $q_0 Z q_1$

2. (i)

$(q_0 Z q_0) \rightarrow a(q_0 A q_0)(q_0 Z q_0) \mid a(q_0 A q_1)(q_1 Z q_0) \mid a(q_0 A q_2)(q_2 Z q_0)$

$(q_0 Z q_1) \rightarrow a(q_0 A q_0)(q_0 Z q_1) \mid a(q_0 A q_1)(q_1 Z q_1) \mid a(q_0 A q_2)(q_2 Z q_1)$

$(q_0 Z q_2) \rightarrow a(q_0 A q_0)(q_0 Z q_2) \mid a(q_0 A q_1)(q_1 Z q_2) \mid a(q_0 A q_2)(q_2 Z q_2)$

(ii)

(iii) $(q_2 A q_0) \rightarrow b$

(iv) $(q_0 A q_1) \rightarrow a$

$\delta(q_i, \epsilon, Z) = (q_j, AB)$?

—— x —— x —— x ——

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

$S \rightarrow aSb \mid \epsilon$

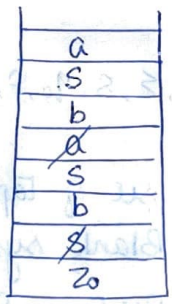
Sample : a a b b

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

$\delta(q_1, \epsilon, S) = (q_1, aSb)$

$\delta(q_1, \epsilon, S) = (q_1, \epsilon)$

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

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

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