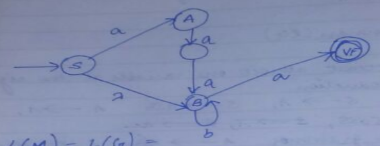
Vini Sawjiani



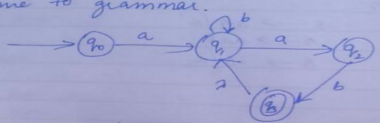
$L(M) = L(G) = aaab^*a + b^*a$

Left linear to right linear:

$L(G')$ $A \longrightarrow \cancel{B} a_1 a_2 \dots a_n$
becomes
$L(G)$ $A \longrightarrow a_1 a_2 \dots a_n B$.
& vice versa. Both the languages ~~are same and~~ regular but are reversed.

$\therefore L(G') \rightarrow L(G')^R = L(G)$

Machine to grammar.



$= ab^*ab(b^*ab)^*$

$\cancel{\frac{q}{q_0}}$

RE for each token $\longrightarrow$ ENFA
combine them into one ENFA by start state
Then ENFA to DFA

Regular grammar (RG)

linear : if ~~atleast~~ atmost one variable at the right side of production.

$S \longrightarrow \lambda, \quad S \longrightarrow aSb \quad A \longrightarrow \lambda,$

Non linear : $S \to SS$, ~~$S \longrightarrow \lambda$, $S \longrightarrow a$~~

Right linear grammar $\quad A \longrightarrow xB$
or
$A \longrightarrow x$

x : string of terminals.
rightmost

left linear grammar : $A \longrightarrow Bx$
or
$A \longrightarrow x$.

RG : left + right linear only
RG generate Reg. languages.

let G be right linear grammar.
let $L(M) = L(G)$
where M is a machine
eg : G :

$S \longrightarrow aA / B$
$A \longrightarrow aaB$
$B \longrightarrow bB / a$