Testing if $L(G) = \phi$.

Given a CFG G we say a NT X is generating if $X \stackrel{*}{\Rightarrow} w$ where $w \in \Sigma^*$ (or T^*).

$\stackrel{*}{\Rightarrow}$ means generate possibly after several steps.

We say X is reachable if $S \stackrel{*}{\Rightarrow} \alpha X \beta$

Thm $L(G) \neq \phi$ iff S is generating.

Algorithm to test if $L(G) = \phi$.

Define GEN to be the set of all generating NTs.

Initialize $GEN = \phi$.

Put all terminal symbols in GEN

Repeat until no more changes

for each rule $X \to \alpha$ do

if every symbol in α is in GEN, put X in GEN

Test if $S \in GEN$.

REMARK : If $A \to \varepsilon$ we put $A \in GEN$.

EXAMPLE 1. $S \to bA \mid aB \quad A \to bAA \mid aS \mid a \quad B \to aBB \mid bS \mid b$

This will yield $GEN = \{a, b, A, B, S\}$ so $L(G) \neq \phi$.

EXAMPLE 2.

$S \to aXb \mid bYa \quad X \to aXb \quad Y \to bYa \quad Z \to ab$

$GEN = \phi$ initially

then $GEN = \{a, b\}$

then $GEN = \{a, b, Z\}$

& there are no more changes.

$S \notin GEN$ at the end so $L(G) = \phi$

There is a similar but more complicated algorithm to test if $L(G)$ is finite or infinite.