INSTITUTE OF INFORMATION TECHNOLOGY

JAHANGIRNAGAR UNIVERSITY

Number of Assignment **:** 04

Submission Date **:** 04/08/2022

Course Tittle **:** Theory of Computing

Course Code **:** ICT - 3105

Submitted To

Dr. Rashed Mazumder

Assistant Professor

IIT – JU

Submitted By

Md. Shakil Hossain

Roll – 2023

3rd year 1st Semester

IIT – JU

Chapter - 7

Answer to the question no - 1

If a language is Context free language then it has

1. Context free grammar (CFG)
2. Push Down automation (PDA)

If the Productions has may or may not Contain 'ε' Production

Example:-

language $L = b^m c^n d^o \mid m, n, o \geq 1$ is a contaxt free language

If does not Produce 'ε' become $m, n, o \geq 1$.

CFG Corespond to CFG

$$\left.\begin{array}{l} A \rightarrow B\ C\ D \\ B \rightarrow b\ B \mid b \\ C \rightarrow c\ C \mid c \\ D \rightarrow d\ D \mid d \end{array}\right\} \text{Does not have '}\varepsilon\text{' Production}$$

let language $L = a$ which is finite and content free language its CFG is

$A \rightarrow a$ Correspond to the above language.

Answer to the question no - 2

A CFG is in Chomsky Normal Form if the Productions are in the following forms

$A \rightarrow a$
$A \rightarrow BC$
$S \rightarrow E$

Where A, B, and C are non-terminals and a is terminal.

Algorithm to Convert into Chomsky Normal Form

Step-1: If the Start Symbol S occurs on Some right side create a new start symbol S and a new Production $S' \rightarrow S$.

Step-2: Remove Null Productions.

Step-3: Remove Unit Productions.

Step-4: Replace each Production $A \rightarrow B_1 \ldots B_n$ where $n > 2$ with $A \rightarrow B_1C$ where $C \rightarrow B_2 \ldots B_n$. Repeat this step for all Productions having two or more symbols in the right side.

Step-5: If the right side of any Production is in the form $A \rightarrow aB$ where a is a terminal and A, B are non-terminal, then the Production is replaced by $A \rightarrow XB$ and $X \rightarrow a$. Repeat this step for every Production which is in the form $A \rightarrow aB$.

Roll - 2023

Answer to the question no - 3

let $G = (V, T, P, S)$ be a CFG and assume that $L(G) \neq 0$.

G generates at least one string.

let $G_1 = (V_1, T_1, P_1, S)$ be the grammar we obtain by the following steps.

Proof: Suppose X is a symbol that remains X is in $V_1 \cup T_1$.

We know that $X \overset{*}{\underset{G}{\Rightarrow}} w$ for some w in T^*.

Moreover every symbol used in the derivation of w from X is also generating.

Thus $X \overset{*}{\underset{G_2}{\Rightarrow}} w$.

Since X was not eliminated in the second step we also know that there are α and β such the $S \overset{*}{\underset{G_2}{\Rightarrow}} \alpha X \beta$.

Further every symbol used in this derivation is reachable so $S \overset{*}{\underset{G_1}{\Rightarrow}} \alpha X \beta$.

We know that every symbol in $\alpha X \beta$ is reachable and we also know that all these symbols are in $V_2 \cup T_2$ so each of them is generating in G_2. The derivation of some terminal string say $\alpha X \beta \overset{*}{\underset{G_2}{\Rightarrow}} xwy$.
involves only symbols that are reachable from S, because they are reached by symbols in $\alpha X \beta$. Thus this derivation is also a derivation of G_1 that is

$$S \overset{*}{\underset{G_1}{\Rightarrow}} \alpha X \beta \overset{*}{\underset{G_1}{\Rightarrow}} x w y$$

We conclude that X is useful in G_1. Since X is an arbitrary symbol of G_1 we conclude that G_1 has no useless symbol.

The last detail is that we must show
$L(G_1) = L(G)$.

As usual to show two sets the same we show each is contained in the orther.

$L(G_1) \subseteq L(G)$

Since we have only eliminated symbols and productions from G to get G_1 it follows that $L(G_1) \subseteq L(G)$.

$L(G) \subseteq L(G_1)$

We must prove that if w is in $L(G)$ then w is in $L(G_1)$. If w is in $L(G)$ the $S \overset{*}{\underset{G}{\Rightarrow}} w$.

Each symbol in this derivation is evidently both reachable and generating so it is also a derivation of G_1.

That is $S \overset{*}{\underset{G_1}{\Rightarrow}} w$ and thus w is in $L(G_1)$.

Chapter - 8

Answer to the question no - 1

A Turing Machine (TM) is a mathematical model which consists of an infinite length tape divided into cells on which input is given. It consists of a head which reads the input tape. A state register stores the state of the Turing machine. After reading an input symbol it is replaced with another symbol its internal state is changed and it moves from one cell to the right or left. If the TM reaches the final state in the input string is accepted otherwise rejected.

A TM can be formally described as a 7-tuple $(Q, X, \Sigma, \sigma, q_0, B, F)$ where

Q is a finite set of states

X is the tape alphabet

Σ is the input alphabet

σ is a transition function

q_0 is the initial state

B is the blank symbol

F is the set of final states.

Example:

Turing Machine $M = (Q, X, \Sigma, \sigma, q_0, B, F)$

$Q = \{q_0, q_1, q_2, q_f\}$

$X = \{a, b\}$

$\Sigma = \{1\}$

$q_0 = \{q_0\}$

B = blank Symbol

$F = \{q_f\}$

THE END