Name: Muhammad Harris

Reg #: BCS203193

Date: 20-12-22

Course: Theory of Automata and Formal Languages

Section: 4

Assignment No: 3

Description: Context-Free Grammar

Submitted to: Sir Hashim Ayub

Capital University of Science & Technology

CUST 2022

Theory of Automata and Formal Languages
Assignment 03

1. Consider the context-free grammar:

$$S \rightarrow aSa \mid bb \mid b$$

prove that this generates the language defined by the regular expression

$$a^* bbb$$

for abbb,

$S \rightarrow aSa$

$S \rightarrow abba$ X

according to the production rule,
P:

$S \rightarrow aSa \mid bb \mid b$

the generated string will always end with 'a', which is against the regular expression a^*bbb. **Hence, above context-free grammar doest not generate language defined by the regular expression a^*bbb.**

2. Consider the following CFG:

$S \to XWG$

$X \to aX \mid bX \mid G$

$W \to bb \mid aa \mid aW \mid bW$

$G \to aG \mid bG$

Prove that this generates the language off all strings with a triple b in them which is the language defined by $(a+b)^* bbb(a+b)^*$.

for bbb,

$S \to XWG$

$S \to XWG$

$S \Rightarrow XbbG$

$S \Rightarrow GbbG$

$S \Rightarrow GbbbG$

Since, G cannot be removed from any string generated by context-free grammar.

$S \to XWG$

$X \to aX \mid bX \mid G$

$W \to bb \mid aa \mid aW \mid bW$

$G \to aG \mid bG$

Therefore, CFG cannot produce strings of language defined by $(a+b)^* bbb(a+b)^*$.

3. Consider the following CFG:

$S \to aX$

$X \to aX \mid bX \mid \Lambda$

What is the language of CFG that will be generated.

The above CFG generates a language that starts with a and has any number of a's and b's.

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

4. What is the language this cfg will be generating while considering the following CFG:

$S \to XaXaX \mid SS \mid \Lambda$

$X \to aX \mid bX \mid \Lambda$

The above CFG generates a language that can be defined by regular expression,

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

5. Consider the CFG:

$S \to SS \mid XaXaX \mid \Lambda$

$X \to bX \mid \Lambda$

Prove that X can generate b^*

for λ	for b	for bbb
$X \to \Lambda$	$X \to bX$	$X \to bX$
	$X \to b\Lambda$	$X \to bbX$
	$X \to b$	$X \to bbbX$
		$X \to bbb\Lambda$
		$X \to bbb$

Proved

6. Show that the set $L = \{a^n b^n \mid n >= 1\}$ is not regular.

By using pumping lemma,

Let $L = \{a^n b^n \mid n >= 1\}$

assuming that L is a regular language. let m be the number of pumping lemma.

Let $S = a^m b^m$

Since $S \in L$ and $|S| \geq m$, the pumping lemma applies specifically,

$S = xyz$ where $y \neq \Lambda$ and $|xy| \leq m$

$y = a^k$ where $0 < k \leq m$

$x = a^q$ where $0 \leq q < m$

$z = a^{m-k-q} b^m$

pumping lemma says that

$xyz \in L$

$xyz = a^m a^q a^k a^{m-k-q} b^m$

$xyz = a^{k+m} b^m \in L$

Thus, our assumption that L is is a regular language is incorrect.

Hence, proved that the set $\{a^n b^n \mid n \geq 1\}$ is not regular.

7. Consider a grammar G is given as follows

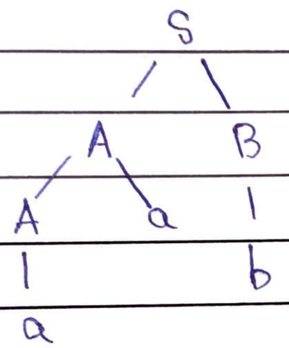
S → AB | aaB

A → a | Aa

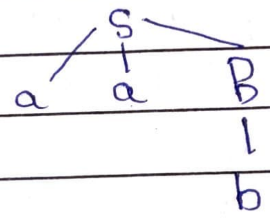
B → b

determine whether grammar G is ambigous or not. If G is ambigous, construct unambigous grammar equivalent to G.

Let us derive for string 'aab'



Parse tree 1



Parse tree 2

As there are two different parse tree for same string, grammar is **Ambigous**

Unambigous Grammar:

S → AB   by using parse tree 1.

A → Aa | a

B → b

8. Simplify the following grammar

$S \rightarrow aAa \mid bBb \mid BB$

$A \rightarrow C$

$B \rightarrow S \mid A$

$C \rightarrow S \mid \varepsilon$

1. Reduction of CFG

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

$W_1 = \{B, C\}$

$W_2 = \{A, B, C\}$

$W_3 = \{S, A, B, C\}$

$W_4 = \{S, A, B, C\}$

modified production rule

$S \rightarrow aAa \mid bBb \mid BB$

$A \rightarrow C$

$B \rightarrow S \mid A$

$C \rightarrow S \mid \varepsilon$

$Y_1 = \{S\}$

$Y_2 = \{S, A, B, a, b\}$ from $S \rightarrow aAa \mid bBb \mid BB$

$Y_3 = \{S, A, C, B, a, b\}$ from $A \rightarrow C$

$Y_4 = \{S, A, C, B, a, b\}$

P:

$S \rightarrow aAa \mid bBb \mid BB$

$A \rightarrow C$

$B \rightarrow S \mid A$

$C \rightarrow S \mid \varepsilon$

2. Removal of unit production:

$S \to aAa \mid bBb \mid BB$
$A \to C$
$B \to S \mid A$
$C \to S \mid \varepsilon$

for $A \to C$,
$A \to S \mid \varepsilon$

removing $C \to S \mid \varepsilon$ from grammar

P:
$S \to aAa \mid bBb \mid BB$
$A \to S \mid \varepsilon$
$B \to S \mid A$

3. Removal of null production:

for $A \to \varepsilon$, new production rule will be

P:
$S \to aAa \mid bBb \mid BB \mid aa \mid bb \mid \varepsilon$
$A \to S$
$B \to S$

Simplified CFG:

$S \to aAa \mid aa \mid bBb \mid bb \mid \varepsilon$
$A \to S$
$B \to S$

$\xrightarrow{\text{further simplification}}$ $S \to aSa \mid bSb \mid aa \mid bb \mid \varepsilon$

9. Discuss in detail about ambigous grammar and removing ambiguity from grammar.

A context free grammar is ambigous if there are more than one parse trees for a given occurance of string that belongs to a language.

Ambiguity can be removed by

1. fixing grammar
2. By ~~only~~ including production rules of a single parse tree that makes most sense.
3. By adding precedence rules.

10. Construct a derivation for the string 0011000 using the grammar

$$S \to A0S \mid 0 \mid SS$$

$$A \to S1A \mid 10$$

0011000

~~S → SS~~

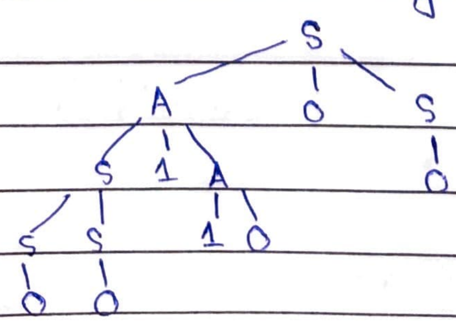
$S \to \cancel{S1}\ A0S$ using $S \to A0S$

$S \to S1A0S$ using $A \to S1A$

$S \to 001A0S$ using $\cancel{A}$ $S \to 00$ by $S \to SS$

$S \to 001A00$ using $S \to 0$

$\boxed{S \to 0011000}$ using $A \to 10$



SPRING 2022