THEORY OF COMPUTATION
MC-304

ASSIGNMENT- II

SUBMITTED BY: AIMAN SIDDIQUA
2K18/MC/008

(1) $S \to 0S1 \mid 0A1$
$A \to 1A0 \mid 10$

In order to obtain a terminal string we need the production $A \to 10$. Also in order to obtain A we have to apply $S \to 0A1$ at some point.
Before applying $S \to 0A1$ we can apply $S \to 0S1$ n-1 times where $n \geq 1$.
After applying $S \to 0A1$ we can apply $A \to 1A0$ m-1 times where $m \geq 1$. finally we apply $A \to 10$ to obtain a terminal string.

$$S \xrightarrow[0S1]{n-1 \text{ times}} 0^{n-1} S 1^{n-1} \xrightarrow{0A1} 0^{n-1} 0A1 1^{n-1} \xrightarrow[1A0]{m-1 \text{ times}}$$

$$0^n 1^{m-1} A 0^{m-1} 1^n \xrightarrow{A \to 10} 0^n 1^m 0^m 1^n$$

Hence the language generated by the grammar with given production rules is:

$$L(G) = 0^n 1^m 0^m 1^n, \quad n, m \geq 1$$

(2) To construct a grammar accepting:

$\{0^n 1^m 0^n \mid m, n \geq 1\} \cup \{0^n 1^m 2^m \mid m, n \geq 1\}$

The required productions are:

$S \rightarrow 0A0 \mid 0B$

$A \rightarrow 0A0 \mid C$

$C \rightarrow 1C \mid 1$

$B \rightarrow 0B \mid D$

$D \rightarrow 1D2 \mid 12$

(3.) A type 2 production is a production of the form $A \rightarrow \alpha$ where $A \in V_N$ and $\alpha \in (V_N \cup \Sigma)^*$. In other words the LHS has no left context or right context.

For example $S \rightarrow Aa$, $A \rightarrow a$, $B \rightarrow abc$, $A \rightarrow \Lambda$ are type 2 production.

A grammar is called a type 2 grammar if it contains only type 2 productions. It is also called a context free grammar.

Let G be a context free Grammar whose productions are: $S \rightarrow S0S1S0S$

$S \rightarrow S0S0S1S$

$S \rightarrow S1S0S0S$

$S \rightarrow \Lambda$

On applying $S \rightarrow \Lambda$ on the first three production we get 010, 001 and 100 respectively.

Also every string in L is of the form $3n$ where $n \geq 1$

Let us assume the result for all strings of length $3n-3$. Let $w \in L$ and let $|w| = 3n$, w will contain one of 010, 001 or 100 as substring. Let that substring be w_1

w can be written as $w_2 w_1 w_3$.

$$S \stackrel{*}{\Rightarrow} w_2 S w_3 \longrightarrow w_2 w_1 w_3$$

Hence by induction its true.

(4.) A production of the form $A \to a$ or $A \to aB$ where $A, B \in V_N$ and $a \in \Sigma$ is called a type 3 production

A grammar is called a type 3 or Regular Grammar if all its productions are type 3 productions. A production $S \to \Lambda$ is also allowed in type 3 grammar but in this case S does not appear on the right hand side of any production.

A regular grammar generating $\{(ab)^n, n \geq 1\}$ consist of the following productions:

$$S \to aS_1, \; S_1 \to bS, \; S \to aS_2, \; S_2 \to b$$

In order to obtain a terminal string we have to apply $S \to aS_2$ and $S_2 \to b$ consecutively. Before that we can apply $S \to aS_1$ and $S_1 \to bS$ any number of times (even zero).

(5.) $S \to 0S1 \mid 0A \mid 0 \mid 1B \mid 1$

$A \to 0A \mid 0$

$B \to 1B \mid 1$

In order to derive 001100, 001010 or 01010 we cannot use $S \to 0S1$ as the first production as it will end in 1 always and we need 0 at the end. $S \to 0$ and $S \to 1$ produce terminal strings so we cannot use those either.

Hence we are left with $S \to 0A$ and $S \to 1B$.
As $A \to 0A \mid 0$ any derivation from this will only give 0^n. Similarly $S \to 1B$, $B \to 1B \mid 1$ will only give 1^n as the terminal strings.

Hence 001100, 001010 and 01010 are not in the language generated by the grammar with given production rules.

(6) (a) $\{a^2, a^5, a^8, \ldots\}$

Ans: $aa(aaa)^*$

(b) $\{a^n \mid n \text{ is divisible by } 2 \text{ or } 3 \text{ or } n=5\}$

Ans: $(aa)^* + (aaa)^* + aaaaa$

Scanned with CamScanner

(c) The set of all strings over $\{a, b\}$ beginning and ending with a.

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

(7.) $L = \{vwv : v, w \in \{a, b\}^*, |v| = 2\}$

The following regular expression represents the above language:

$$aa(a+b)^*aa + ab(a+b)^*ab + ba(a+b)^*ba + bb(a+b)^*bb$$

Hence the language is regular.

(8.) $L = \{w \in \{0,1\}^* : w \text{ has no pair of consecutive zeroes}\}$

Regular expression: $(01 1^*)^*$

(9.) To prove:

$$(a^*ab + ba)^* a^* = (a + ab + ba)^*$$

LHS = $(a^*ab + ba)^* a^*$

Let $r1 = (a^*ab + ba)^*$ and $r2 = a^*$

Keeping $a^* = \epsilon$ we obtain the LHS term as $(ab+ba)^*$

Scanned with CamScanner

Now let $r1 = \epsilon$, so $r1^* = \epsilon$, hence LHS term is reduced to a^*

Hence the overall expression considering the above 2 cases can be briefed as : $(a+ab+ba)^*$ as $(ab+ba)^*$ is already obtained and using a^* we can get any combination of a so it can be inside $(a+ab+ba)^*$

Hence the two expressions given are equivalent.

10. $L = \{awa : w \in \{a,b\}^*\}$ is

We can construct the following regular expression for the given language :

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

Hence the language is regular.

Scanned with CamScanner