CHAPTER 2 LANGUAGES

In what follows, we write A^* to represent the set of all finite sequences or strings formed using the symbols in A.

eg(1)

If A = {0,1}, some elements of A* are:

eg(2)

If A = {he, runs, slowly}, some elements of A* are:

A <u>language</u> consists of these components:

(1) S = set of all symbols or "words" that are allowed.

(2) Syntax of the language - a set of rules on how the words in S can be arranged to form acceptable sentences, ie what sequences of words are considered "properly constructed sentences".

(3) Semantics of the language - specification of the meaning of properly constructed sentences.

Not all strings in S* are properly constructed sentences.
Not all properly constructed sentences have meaning.

eg(3)

In English, the sequence of words _______________ is not a properly constructed sentence.

_______________________, __________________________are properly constructed sentences, but has no meaning.

eg(4)

In the language of elementary arithmetic, the symbol set is
$S = \{0,1,2,3,4,5,6,7,8,9,+,-,\times,\div,(,)\}$.
These symbols can be arranged in some proper order to form "syntactically correct" arithmetic expressions.

$((3 + 4) \times 0 - 4)$ is a properly constructed sentence.
The meaning is ____

$((3 + \times 4) \times - 0 - 4$ is ___________________________

$3 \div (2 - 2)$ is ____

Phrase Structure Grammar

A phrase structure grammar consists of these components:

(1) S = set of all allowed "words". These words are used to form sentences.

(2) V = a finite set consisting of all the words in S and some additional symbols used to describe the structure of sentences but do not become parts of sentences.

(3) A production relation, $\mapsto$ representing a set of substitution rules specifying what strings may be replaced by some other strings in constructing sentences.

eg. w $\mapsto$ z means the string w may be replaced by the string z whenever w occurs.

$v_o \in V - S$, representing the starting point for substitution.

Eg:

too **fast** - Adverb phrase (AdvP)
very **happy** - Adjective phrase (AP)
the massive **dinosaur** - Noun phrase (NP)
at dinner - Preposition phrase (PP)
watch movie - Verb phrase (VP)

eg(5)

Consider the phrase structure grammar $G = [V, S, v_o, \mapsto]$ where

$S = \{He, She, drives, runs, carefully, slowly, frequently\}$

$V = S \cup \{sentence, noun, predicate, verb, adverb\}$

v_o = sentence

Sentence $\mapsto$ noun predicate

noun $\mapsto$ He

noun $\mapsto$ She

predicate $\mapsto$ verb adverb

verb $\mapsto$ drives

verb $\mapsto$ runs

adverb $\mapsto$ carefully

adverb $\mapsto$ slowly

adverb $\mapsto$ frequently

Show how the sentence "He drives carefully" may be constructed.

Solution:

$$
\begin{aligned}
\text{Sentence} &\mapsto \text{noun predicate}\\
&\mapsto \text{noun verb adverb}\\
&\mapsto \text{He verb adverb}\\
&\mapsto \text{He drives adverb}\\
&\mapsto \text{He drives carefully}
\end{aligned}
$$

The process of substitution may be represented in a tree diagram:

This is called a derivation tree for the sentence

____

The words in S are called **terminal symbols**. Any sentence constructed must consist of terminal symbols only. The symbols in V−S, such as "sentence", "noun", "predicate", "verb", and "adverb" are non-terminal symbols. They should be totally replaced by terminal symbols in the final sentence produced.

Eg: integer $\mapsto$ digit

digit $\mapsto$ 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

____________________________ are terminal symbols.

__________________________are nonterminal symbols.

Direct derivability

y is said to be directly derivable from x if x can be substituted by y using one of the productions to replace all or part of x.

Then we write $x \Rightarrow y$.

The relation $\Rightarrow$ is called direct derivability.

eg(6)

In the last example,

______________is directly derivable from _____________

_______is directly derivable from __________

eg(7)

Suppose a production rule is: $aV \mapsto bcd$

Then baVe may be replaced by __________

bbcde is directly derivable from _________

The transitive closure of $\Rightarrow$ is written as $\Rightarrow^{\infty}$

$x \Rightarrow^{\infty} y$ means x can be converted into y by using a sequence of substitutions, ie by using a combination of one or more productions. We say y is derivable from x.

eg(8)
In eg(5), show that
sentence $\Rightarrow^{\infty}$ He verb adverb
sentence $\Rightarrow^{\infty}$ He drives carefully.

Solution:

If w $\in$ S* and $v_0 \Rightarrow^{\infty}$ w, then w is called a

[w $\in$ S* means w is a string formed using terminal symbols only.
$v_0 \Rightarrow^{\infty}$ w means w is derivable from v_0 by a sequence of substitutions.]

Let G = [V, S, v_0 , $\mapsto$] be a Grammar. The set of all properly constructed sentences is called the language of G.
L(G) = {w: w $\in$ S* and $v_0 \Rightarrow^{\infty}$ w }

eg(9)

Consider the grammar in eg(5).

"He drives carefully" $\in$ L(G).

How many properly constructed sentences are there in L(G) ?

Types of Grammars

Grammars are partitioned into 4 types, based on their complexity. Let G = (V, S, v_o, $\mapsto$) be a phrase structure grammar. Then we say G is

Type 0: If no restrictions are placed on the productions of G

Type 1: I f for any production $w_1 \mapsto w_2$, the length of w_1 is less than or equal to the length of w_2 (where the length of a string is the number of words in that string)

Eg:

Type 2: If the left side of each rule is a single nonterminal and the right -hand side has one or more symbols. Type 2 grammars are called context- free grammars since a nonterminal symbol that is the left side of a production can be replaced in a string whenever it occurs, no matter what else is in the string.

Eg:

Type 3: If the left -hand side of each production is a single, nonterminal symbol and right- hand side has one or more symbols, including at most one nonterminal symbol, which must be at the extreme right of the string.

Eg:

Types of Grammars

BNF notation (Backus-Naur Form)

Production relation $\mapsto$ is represented as ::=

Non-terminal symbols are put in < >

Symbols without < > are terminal symbols.

Multiple alternatives are separated by |

eg(10)

Rewrite the production relation in eg(5) in BNF notation.

Solution:

eg(11)

Let G = [V, S, v_0, $\mapsto$] be given by V = {v_0, w, a, b, c}, S = {a, b, c}.

Production $\mapsto$ described in BNF notation:

<v_0> ::= a<w>

<w> ::= bb<w>|c

Sketch a tree diagram to show how sentences may be produced.

Solution:

A syntactically correct sentence is one of these:

So $L(G) = \{ab^{2n}c$ where $n =$ an integer $\geq 0\}$
May also write $L(G) = \{a(b^2)^*c\}$
where * means "repeat any number of times".
A production in which the symbol on the left occurs again on the right, such as <w> ::= (...)<w>, is described as a <u>recursive production</u>. Such a production can be used repeatedly any number of times.

<u>Parsing a sentence</u>

Given a sentence, we wish to check whether it is syntactically correct. It is analysed to show the structure, and a derivation tree may be drawn for it. This process of analysing sentence structure is called <u>parsing</u>, and the tree obtained is called a <u>parse tree</u>.
In converting a programming language, a compiler parses a sentence and searches the parse tree. As each vertex is visited, the sentence is translated into another language.

eg(12)

For the language of the Grammar in eg(5), obtain a parse tree for the sentence "She runs slowly".

<u>Syntax diagrams</u>

These diagrams are drawn to represent productions (substitution rules). Terminal symbols are put in circles. Non-terminal symbols are put in square boxes. Arrows show direction of flow.

<u>eg(13)</u>

<v_o> ::= a<w> is drawn as

Solution:

<w> ::= a|b|c<x>

Solution:

<w> ::= <x><y>|<x>a|bc<y> is drawn as

The recursive production <w> ::= ab<w> may be drawn as

<w> ::= bb<w>|c is drawn as

<w> ::= ab | ab<w> may be drawn as

If we can combine all the component diagrams into a single diagram which involves only terminal symbols, that diagram is called the master diagram of that Grammar.

eg(14)

Draw the master diagram for $G = [V, S, v_0, \mapsto]$ where $S = \{a,b,c\}$, $V = S \cup N$, $N = V - S = \{v_0, w\}$.

$<v_0> ::= a<w>$

$<w> ::= bb<w>|c$

Solution:

Reading this diagram following the arrows, we obtain properly constructed sentences like:

____

eg(15)

Let the set of terminal symbols S = {0,1,2,3,4,5,6,7,8,9,•}

Set of non-terminal symbols N = {decimal number, decimal fraction, unsigned integer, digit}.

$V = S \cup N$

v_o = decimal number

<decimal number> ::= <unsigned integer> | <decimal fraction>|
<unsigned integer><decimal fraction>

<decimal fraction> ::= •<unsigned integer>

<unsigned integer> ::=<digit>|<digit><unsigned integer>

<digit> ::= 0|1|2|3|4|5|6|7|8|9

Draw a syntax diagram for this Grammar.

Draw a derivation tree for the decimal number 56.84

Is 56.84.12 a syntactically correct decimal number ?

Solution:

$\boxed{d}$ — represents the above.

Derivation Tree for 56.84.

Reading through the master syntax diagram, we see that a decimal number has at most one decimal point whichever branch we follow, so 56.84.12 is not syntactically correct.

Regular Grammars & Regular Expressions

A regular grammar is a grammar in which the left-hand side of each production is a single, nonterminal symbol, and the right-hand side has one or more symbols including at most one nonterminal symbol, which must be at the extreme right of the string.

eg:

The language of a regular grammar can be represented by a regular expression. A regular expression over the set S is a string formed using the symbols in S, possibly with the help of the symbols * , $\vee$, and ( ), where

a* means a repeated any number of times = a^n for $n \geq 0$,
$a \vee b$ means either a or b, that is $a^n b^{1-n}$ for n = 0 or 1.
( ) may be used to enclose a sequence of symbols treated together as one block.

Example:

ab* represents strings like ______________________________

(ab)* represents strings like _____________________________

a v b* represents _____________________________________

(a v b)* = ____

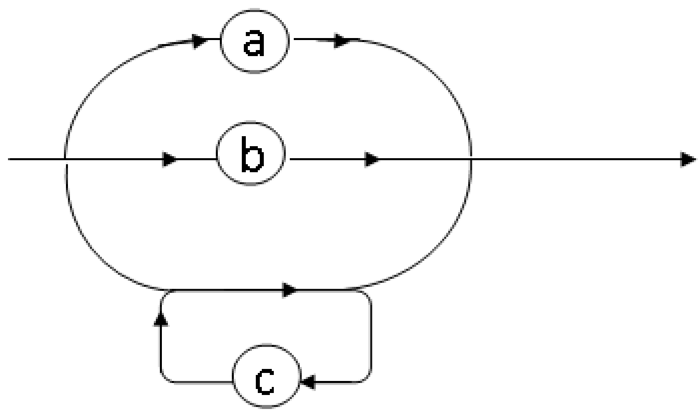
Strings represented include ____________________________

$(ab)^n$ for $n \geq 1$ may be written as _______________________

There is a correspondence between segments of the master diagram of a Grammar and regular expressions. Alternative branches correspond to $\vee$. A loop corresponds to *. Segments in sequence correspond to strings joined together.

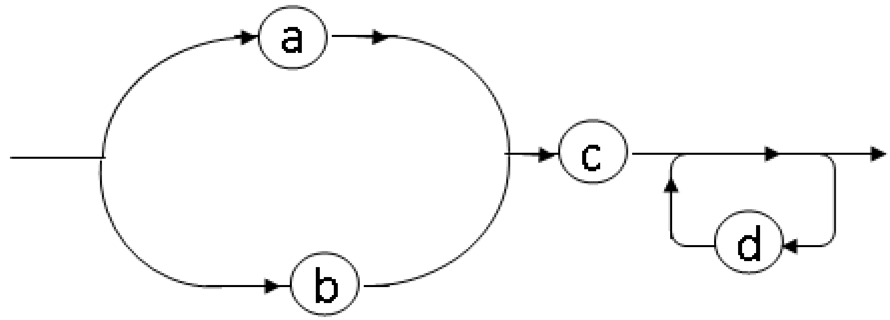
Ex.1: Write a regular expression to represent the language of each Grammar whose master diagram is given as follows:

(a)



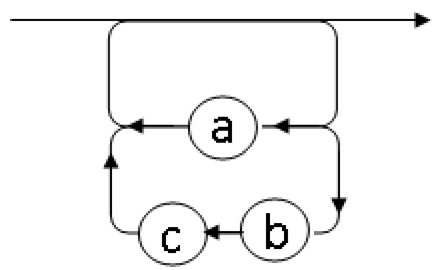
Solution:

(b)



Solution:

(c)



Solution:

Ex.2 Draw a master diagram for each of the Grammars whose languages are represented by these regular expressions:

(1) $(abc) \vee (de)$

Solution:

(2) $ab(c \vee d)e$

Solution:

(3) abc^*d

<u>Solution:</u>

(4) $a(bc)^*d(e \vee f)$

<u>Solution:</u>

(5) $abc(bc)^*d(e \vee f)^*g$

OR

Extra Example 1:

A Grammar, G is described in Backus-Naur Form (BNF) by
$< v_0 > ::= ab< v_o > \mid ac< v_o > \mid bb< v_1 >$
$< v_1 > ::= bb< v_1 > \mid c< v_2 > \mid d< v_2 >$
$< v_2 > ::= e< v_2 > \mid e$

(i) List all the nonterminal symbols used in G.

(ii) List all the terminal symbols used in G.

(iii) Draw the syntax diagram for G.

Extra Example 2:

Given G = (V, S, v_0, $\rightarrow$) is a grammar where V = { v_0, v_1, v_2, x, y, z }, S = {x, y, z} and the production relation $\rightarrow$ described by

$$
\begin{aligned}
v_0 &\rightarrow xyv_0 \\
v_0 &\rightarrow yv_1 \\
v_1 &\rightarrow yv_1 \\
v_1 &\rightarrow zzv_2 \\
v_1 &\rightarrow xv_2 \\
v_2 &\rightarrow y
\end{aligned}
$$

(i) Write the BNF (Backus-Naur Form) for the production.

(ii) Draw the master syntax diagram for G.

(iii) Draw a derivation tree for $xyyzzy$.

(iv) Write the L(G).

Extra Example 3:

Let G = (V, S, v_0,) be a grammar where V = {v_0, v_1, v_2, a, b}, S = {a, b} and the production relation $\longmapsto$ is described as:

$v_0 \longmapsto bav_1$
$v_1 \longmapsto av_0$
$v_1 \longmapsto bv_2$
$v_1 \longmapsto b$
$v_2 \longmapsto bbv_2$
$v_2 \longmapsto babv_2$
$v_2 \longmapsto a$

(i) Write the Backus-Naur Form (BNF) for the production.
(ii) Draw the master syntax diagram of G.
(iii) Draw the derivation tree for the sentence babbbbaba.
(iv) Write the regular expression to represent the form of all possible syntactically correct sentences.

Extra Example 4:

Let $G = (V, S, v_o, \mapsto)$ where $V = \{v_o, v_1, v_2, 0, 1\}$, $S = \{0, 1\}$ and the production relation $\mapsto$ is described as follows:

$v_0 \mapsto 1v_0$
$v_0 \mapsto 101v_1$
$v_1 \mapsto 00v_1$
$v_1 \mapsto 1v_2$
$v_1 \mapsto 11$
$v_2 \mapsto 1v_2$
$v_2 \mapsto 0v_2$
$v_2 \mapsto 01$

(i) Write the Backus-Naur Form (BNF) for the production.

(ii) Draw the master syntax diagram to illustrate the production of G.

(iii) Write the regular expression that corresponds to the master syntax diagram in part (ii).

(iv) Draw the derivation tree for the sentence 110110001.

Extra Example 5:

G = (V, S, v_0, $\mapsto$) is a grammar, where V = { v_0, v_1, v_2, a, b, c, d, e, f, g }, S = { a, b, c, d, e, f, g } and the production relation $\mapsto$ is described as follows:

$v_0 \mapsto a\, v_0$
$v_0 \mapsto b\, v_0$
$v_0 \mapsto c\, v_1$
$v_1 \mapsto d\, d\, v_1$
$v_1 \mapsto e\, e\, v_2$
$v_1 \mapsto f\, v_2$
$v_2 \mapsto g$

(i) Write the production relation in BNF notation.

(ii) Draw a derivation tree for the sentence *abacddfg*.

(iii) Determine whether *aacdddeeg* is a synthetically correct sentence.

(iv) Draw the master syntax diagram for G.

(v) Write a regular expression to represent the form of all possible synthetically correct sentences.

Extra Example 6:

Let $G = (V, S, v_o, \longmapsto)$ where $V = \{v_o, v_1, v_2, a, b, c, d\}$, $S = \{a, b, c, d\}$ and the production relation $\longmapsto$ is described as follows:

$$
\begin{aligned}
v_o &\longmapsto v_1 \\
v_o &\longmapsto v_2 \\
v_1 &\longmapsto abv_1 \\
v_1 &\longmapsto abv_3 \\
v_1 &\longmapsto d \\
v_2 &\longmapsto abbv_2 \\
v_2 &\longmapsto d \\
v_3 &\longmapsto ccv_3 \\
v_3 &\longmapsto d
\end{aligned}
$$

(i) Write the Backus-Naur Form (BNF) for the production.

(ii) Draw a master syntax diagram of G.

(iii) Find the regular expression that corresponds to the master syntax diagram.

(iv) Draw a derivation tree for the sentence *ababccccd*.