Context Free Grammar (CFG) for MODI Script

Shubhangi Bhatambrekar, and Niket Tajne

Abstract**—**The MODI script is one of the cursive type of writing in Marathi which is the primary language spoken in the state of Maharashtra in western part of India. The MODI word originated from the word “MODANE” meaning “to break”. MODI Script can be used as a shorthand script for faster writing in business and other administrative work. MODI language grammar is similar to Marathi language grammar. When we write any grammatically sentence of a language in computational forms, initially it is quite difficult to arrange them. This paper shows how we can write context free grammar for MODI script since Context Free Grammar (CFG) is suitable for representing the natural language properly in computational form. Also CFG is helpful in pattern recognition for the valuable script. This paper also highlights the process of specifying CFG for simple MODI Script sentences.

Keywords**—**Context Free Grammar (CFG), MODI script, Pattern Recognition, Parsing.

I. Introduction

IN many languages like English, Japanese, Hindi, etc. the words can be formed by a several ways naturally. Some formations are constructing for the valid sentences and some for invalid sentences. The validity of the sentence is checked or determined by the grammar of a language i.e. Grammar is the set of rules. Grammar is also Collection of Rules that describes valid sentences in a Language[2].

Context Free Grammar for the Naturallanguage is the most important area for the research from many years. That’s way the Context Free Grammar (CFG) for the various Indian as well as foreign languages are available. The MODI script has got the importance because of its effective use in business and other administrative work [5] because of its fast writing of the characters. That is why some times the MODI script is called as Short-Hand writing.

In this paper we tried to write the Context Free Grammar for the MODI script. Section 2 describes Introduction of CFG, Section 3 Describes MODI Context Free Grammar; Section 4 describes the Parser and its Types and finally Section 5 Concludes the paper.

II. Context Free Grammar

Context Free Grammar is a simple recursive method of specifying grammar rules by which string in a language can be generates. Context free grammar is also called as the Type-

2 grammar [1]. This class or type of grammar generates a large and rich class of languages which are suitable for machine communications.

There are four important terms related to the CFG are Variables, Terminals, Production and Start symbol. Where the language represented by the variables, is described in terms of primitive symbols called terminals and set of rules related to the variables are called production and each production start from particular position called start symbols i.e. Context free grammar is collection of four tuples[1],[2],[4].

G = (V, T, P, S)

Where,

V = A finite set of Variables or Non-Terminals

T = A finite set of Terminals

P = A finite set of Production Rules

S = Start symbol i.e. S Є V

If grammar is in A → β[4]where A Є V and β is string of terminals and variable or non-terminals then only we said that the given grammar is in CFG otherwise such a grammar is called context sensitive grammar[1]. The context free grammar the most important phase of the grammar is parser i.e. reading the each and every character of the string and that small character called the token, while reading the string the tree structure is created and it is called a parse tree or syntax tree. The main goal of parse tree is to show the hierarchical structure of the language [3]. In which root node and intermediate nodes contain the variables or non-terminals and leaf node contain always terminals [1],[2]. Parsing plays an important role while performing many computational tasks [1].

For example

$L = \{ a^n b^n \mid n\geq1\}$

is the implicit definition of the language which accept the equal no of a’s and b’s which start with a and end with b always like

L = {ab, aabb, aaabbb,………..}

So the grammar is

S → aSb | ab

So we can accept ab like S → ab then next string we can construct aabb sentential form

S → aSb S → aabb

Dr. Shubhangi Bhatambrekar is Associate Professor, Modern College of Arts, Science & Commerce, Ganeshkhind, Savitribai Phule Pune University, India.

Prof. Niket Tajne is Assistant Professor, Modern College of Arts, Science & Commerce, Ganeshkhind, Savitribai Phule Pune University, India.

http://dx.doi.org/10.15242/IIE.E0415011

Here replacment of S as anon-terminal value is depending on the sentential form.

III. MODI CONTEXT FREE GRAMMAR

MODI is a Brahmi-based script used for writing Marathi language which is Indo-Aryan language used in Western India, because of the speed of writing in administrative affairs [5],[7]. MODI script used in Royal Families of the Indian State Maharashtra. The MODI word is translation of the Persian word “Shikasta” Which means “breaking” in English [5]. As compare with the Marathi language MODI script is simplest version of it because of the breaking the character of the word and same methodology apply in the Parsing of the sentences in Context Free Grammar. The MODI alphabet are variant of the Marathi Language and it has 14 vowels and 34 consonant letters [6],[7] as shown in Figures 1 & 2.

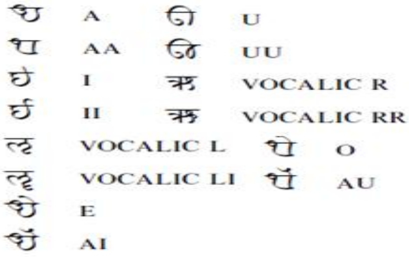

A
AA
I
II
VOCALIC L
VOCALIC LI
E
AI
U
UU
VOCALIC R
VOCALIC RR
O
AU

Fig 1 MODI Vowels

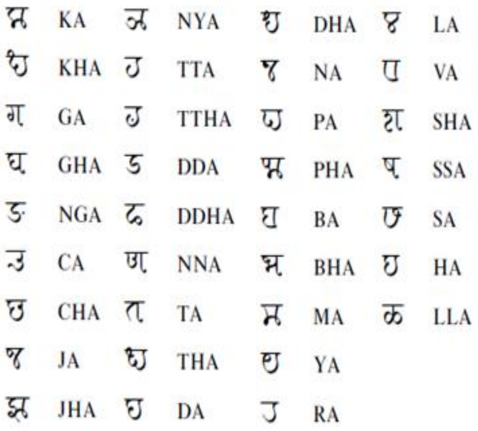

KA
KHA
GA
GHA
NGA
CA
CHA
JA
JHA
NYA
TTA
TTHA
DDA
DDHA
NNA
TA
THA
DA
DHA
NA
PA
PHA
BA
BHA
MA
YA
RA
LA
VA
SHA
SSA
SA
HA
LLA

Fig 2 MODI Consonants

A. Cfg For Simple Modi Sentence

Creating grammar for whole MODI script sentences is difficult; So we select a simple MODI script sentence and create the Context Free Grammar for each of them. This paper presents Context free grammar with simple MODI grammatical productions which will generate a small set of MODI sentence[1].

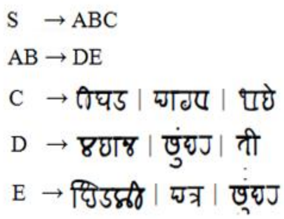
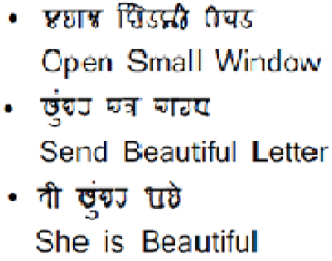


Fig. 3 MODI Grammatical Productions

Fig 3 Shows the MODI Grammatical Productions for simple instructions like in English language [1],[2].


Open Small Window
Send Beautiful Letter
She is Beautiful

While translating the above three sentences in to the Context free grammar, we need to read each and every character from the sentences. This reading method is called the Parsing.

IV. PARSERS

The Parser / Syntax analyzer basically checks for the syntax of the language [1]. A syntax analyzer takes the tokens form the Context free grammar and Sentences and groups them in such a way that some syntax can be recognized. After grouping the tokens if at all any syntax cannot be recognized then error will be generated.

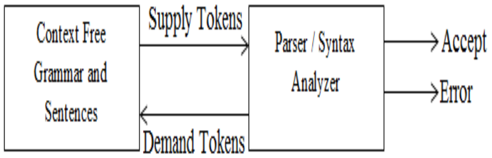


Fig 4 Parser / Syntax Analyzer

Figure 4.1 shows the working of parser with context free grammar. It accepts sentences or it gives the syntactical error. Parsers are of two kinds: Top-Down Parser and Bottom-Up Parser [1].

A. Top-Down Parser

The Top-Down Parsing builds a set of rooted candidate parser trees from left to right over the string [8]. When the parse tree can be constructed from root and expanded to leaves then such type of parser is called Top-Down Parser. The name itself tells us that the parse tree can be built from top to bottom. The main task in Top-Down parsing is to find the appropriate production rule in order to produce the correct input string [1]. We will understand the process of Top-Down parsing with the help of grammatical production example.

For Example

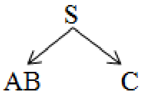
http://dx.doi.org/10.15242/IIE.E0415011

S	→	ABC
AB	→	DE | DEF
D	→	𑘦𑘰𑘖𑘿𑘧𑘰 | 𑘝𑘰𑘕𑘲
E	→	𑘁𑘃 | 𑘝𑘱𑘝𑘱
F	→	𑘄𑘟𑘿𑘧𑘰
C	→	𑘑𑘬𑘢 | 𑘬𑘿𑘡𑘰

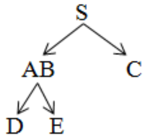
For the above Grammatical Production we accept one input sentence and construct the parse tree for that input sentence by using the Top-Down Parsing method.

Step 1 Start building the sentence from start symbol i.e. S

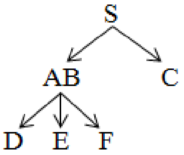
Step 2 Insert the Alternative for the S i.e. S → ABC



Step 3 Insert the first Alternative for the AB i.e. AB → DE

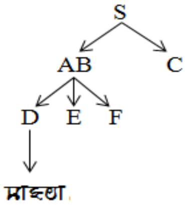


Step 4 It is not satisfied are sentence condition thus we Insert the Second Alternative for the AB i.e. AB → DEF



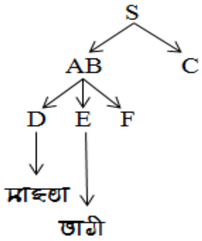
Step 5 We Insert the First Alternative for the D i.e.

D → 𑘦𑘰𑘖𑘿𑘧𑘰



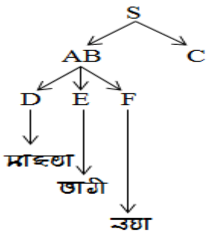
Step 6 We Insert the First Alternative for the E i.e.

E → 𑘁𑘃



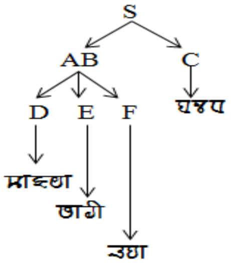
Step 7 We Insert the Alternative for the F i.e.

F → 𑘄𑘟𑘿𑘧𑘰



Step 8 Finally we insert the first Alternative for the C i.e.

C → 𑘑𑘬𑘢



B.Bottom Up Parser

The parse tree can be constructed from leaves to root such type of parser is called as Bottom-Up Parser [2]. Thus the parse tree is built in bottom up manner. In this process, the input symbols are placed at the leaf nodes after successful parsing. The Bottom-Up Parse tree is created starting from leaves, the leaf nodes together are reduced further to internal nodes, these internal nodes are further reduced and eventually a root node is obtained.

The Bottom-Up Parse tree construction process indicates that the tracing of derivations are to be done in reverse order. In the Bottom-Up Parser two fundamental operation are used these are Shift and Reduce these operations are similar to PUSH and POP operation of the stack[2]. Following Table I. describes the process of shifting and reducing the input string in to stack of the above grammatical production.

TABLE I.
BOTTOM UP PARSERR PROCEDURE

Steps	Stack	Parsing Action
I	$	𑘝𑘰𑘕𑘲 𑘝𑘱𑘝𑘱 𑘬𑘿𑘡𑘰
II	$ 𑘝𑘰𑘕𑘲	Reduce (D → 𑘝𑘰𑘕𑘲)
III	$ D	𑘝𑘱𑘝𑘱 𑘬𑘿𑘡𑘰 Shift
IV	$ D 𑘝𑘱𑘝𑘱	Reduce (E → 𑘝𑘱𑘝𑘱)
V	$ D E	Reduce (AB → DE)
VI	$ AB	𑘬𑘿𑘡𑘰 Shift
VII	$ AB 𑘬𑘿𑘡𑘰	Reduce (C → 𑘬𑘿𑘡𑘰)
VII	$ AB C	Reduce (S → ABC)
IX	$ S	Accept

V. CONCLUSION

In this paper we have shown Context Free Grammar (CFG) for MODI script since Context Free Grammar (CFG) is suitable for representing the natural language properly in computational form. Also CFG is helpful in pattern recognition for the valuable script. This paper also highlights the process of specifying CFG for simple MODI Script sentences with examples.

http://dx.doi.org/10.15242/IIE.E0415011

REFERENCES

[1] Anshuman, Pandey, 2011. Preliminary Proposal to Encode the Modi Script" ISO/IEC 10646.

[2] B.M Sagar, Dr.Shobha GI, DrRamakanth Kumar P2. Context Free Grammar Analysis for simple Kannada Sentences.

[3] Carey, William 1810 A dictionary of the Mahratta language . Serampore: Serampore Missionary Press.

[4] DhanashreeKulkarni, Specifying Context Free Grammar for Marathi Sentence

[5] D. N. Besekar, Special Approach for Recognition of Handwritten MODI

[6] N. Chomsky, Three models for the description of language,IRE Trans. Info.Theory 2 (3) (1956), 113-124. http://dx.doi.org/10.1109/TIT.1956.1056813

[7] Rakesh A. Ramraje, History of MODI Script in Maharashtra.

[8] Roark B. Probabilistic Top–Down Parsing and Language Modeling,Association for Computational Linguist, 2001

[9] ShihadehAlqrainy, HasanMuaidi, Mahmud S. Alkoffash. Context Free Grammar Analysis for Arabic Sentences.

AUTHOR BIOGRAPHY

Shubhangi Bhatambrekar has qualifications MSc(Physics), MPhil(Physics), MCA, PhD(Computer Science), DCL.She is working as Associate Professor & Head of Computer Science Department at Modern College, Ganeshkhind, Savitribai Phule Pune University, India since 1998. Her areas of interest include Theory of Computation, Data Management System, Software Designing and Software Testing.

Niket Tajne received his BCS (Bachelor of Computer Science) degree in the year 2008 and he has completed MCA (Master of Computer Application) in the year 2011 from Savitribai Phule Pune University, India. He is working as Assistant Professor in the Department of Computer Science of P.E societies Modern College of ASC , Ganeshkhind, Savitribai Phule Pune University, India since 2012. His areas of interest include Theory of Computation, Compiler Design, software designing and Software Testing.