Specifying Context free Grammar for Marathi Sentences

Dhanashree Kulkarni
Assistant Professor
Dr.D.Y.Patil College Of Engineering , Ambi

ABSTRACT
Marathi is an Indo-Aryan Language and forms the official language of state of Maharashtra. It is ranked as the 4th most spoken language in India and 15th most spoken language in the world. When Computational Linguistic is concerned, writing grammar production for a language is a bit difficult because of different gender and number forms. This paper is an effort to write context free grammar for Marathi sentences. CFGs are very much suitable in expressing natural languages as well as programming languages and hence form a major part in field of natural language processing and pattern recognition. This paper highlights the process of specifying CFG for simple Marathi sentences

General Terms
Context Free Grammar, syntactical analysis, Pattern Recognition, linguistics, Parsing, Natural Language Processing

Keywords
Context free grammar, Marathi sentence.

1. INTRODUCTION
Linguistics refers to the study of Languages. The topic of interest of Linguistic researchers is mainly formalization of natural Languages. Natural Language processing is an emerging field of research exploring how a computer can be used to do useful work by making it understand and manipulate natural language text and speech [5]. NLP can be defined as a computerized approach involving set of theories and technologies to analyze text.

In linguistics, the description of a language is split into two parts [1]. One is the grammar consisting of rules describing correct sentence formation. Another is lexicon listing words and phrases that can be used in sentences. In linguistics the relevant constraints of communicative situation that influence language use, language variation and discourse is referred to as context. In the study of formal languages, the surrounding text, that is, what appears before and what appears after the current symbol in the string is referred to as context [2]. Context free means being able to take an action irrespective of the context, that is, irrespective of the surroundings. Same applies to Context free Grammar (CFG). The productions in CFG are such that their left hand sides are free of any context. Such grammars are used as a mathematical system for modeling structure in Natural Language [4].

In the field of Natural Language Processing, a special area of focus has been Asian languages [5]. Substantial work has been done in the case of Hindi. This paper focuses on Marathi – a language spoken by over 70 million people.

Here is an attempt to write context free grammar (CFG) for simple Marathi sentences. Section II describes the Context Free Grammar, Section III refers to Marathi CFG, Section IV is about the parsers and Section V concludes the paper.

2. CONTEXT FREE GRAMMAR
Context Free Grammar is known to be a popular grammar generation method which is extensively used to define syntax of languages [8]. Context free grammar is also called as Type 2 grammar or phrase structured grammar. A CFG is a quadruple having following four components:

1. A set of variables called non-terminals that denote sets of strings.
2. A set of tokens called terminals which are basic symbols from which strings are formed.
3. A set of production rules.
4. A start symbol.

Non-terminals specify a form of hierarchical structure on the language that is important in case of syntax analysis and translation [2]. They are used as intermediate quantities in the generation of outcome consisting solely of terminal symbols. The productions of a grammar specify how terminals and non-terminals can be combined to form strings. It is this set of productions along with terminal symbols that principally gives the grammar its structure.

Formally, grammar G is denoted as G= (V, T, P, S) where V is set of variables, T is set of Terminals, P is set of Productions and S is start symbol [8]. A grammar is said to be a context free grammar if all productions are of the form

A - > h where h belongs to (VUT)* and A is a non-terminal.

Parsing is a very important part of many disciplines of computer science [5]. For example, compilers must parse source code to be able to translate it into object code. In the same way, any application that processes complex commands must be able to parse the commands. In CFGs , the application of a production rule for parsing is independent of the context in the sentential form where it is applied. For example: If there is a production S - > aSb and a sentential form aaSbb then it is possible to apply the production rule to the sentential form to derive the next sentential form aaaSbbb and the action is said to be context free. Replacement of S by aSb was done without looking at what was to the left or right of S in the sentential form. If all productions are of this nature, then the grammar is a context free grammar [2]. A grammar that is not context free is context sensitive grammar.

3. MARATHI CFG

Marathi is an Indo-Aryan Language and forms the official language of state of Maharashtra. Marathi comes under Devanagari script [6]. It is the most popular script in India and has 12 vowels and 35 consonants as shown in Fig 1:

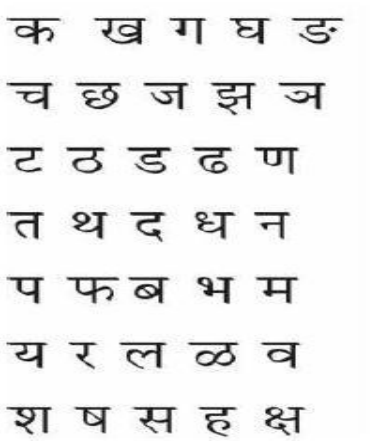


Fig 1. Marathi Consonants

3.1 Specifying CFG for simple Marathi sentences

It is very difficult to design a grammar for the entire Marathi language. For the sake of simplicity, this paper considers writing very simple grammatical productions with CFG that will generate a subset of Marathi sentences [1].

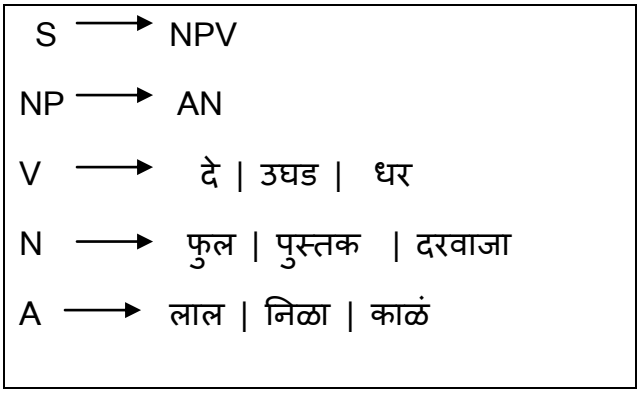


Fig 2 . Marathi Grammatical Production

Fig.2 shows the Marathi Grammatical productions for a Robot to explain simple instructions like [1]:

- निळा दरवाजा उघड

  Open blue door
- काळं पुस्तक धर

  Hold black book
- लाल फुल दे

  Give red flower

Interpretation of the grammar in Fig 2. requires parsing. Top-Down and Bottom-up are two types of parsers that can be used to parse the given grammar[4]. A tree that represents the syntactic structure of a string according to some formal grammar is called a parse tree. In a parse tree, the interior nodes are labeled by non –terminals of the grammar, while the leaf nodes are labeled by terminals of the grammar [1]. Parse tree for above sentences is given in the following section

4. PARSERS

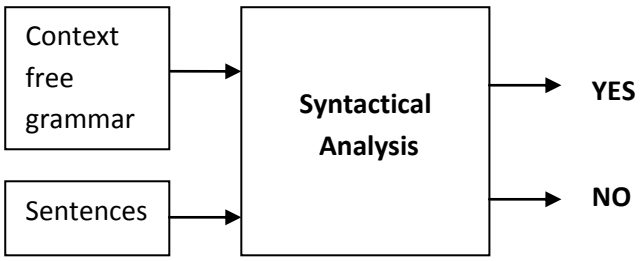


Fig 3. Syntactical analysis

Parsing or syntactic analysis is the process of analyzing a string of symbols, according to the rules of a formal grammar as shown in Figure 6. It is the task of analyzing the grammatical structure of natural language [11].

A parser forms separate units like subject, verb, and object and determines the relations between these units. The linguistic rules provided are mostly in context-free grammar form. This form basically provides a simple and mathematically precise mechanism for describing the methods by which phrases in some natural language are built from smaller blocks [3]. It also gives the basic recursive structure of sentences, the way in which clauses nest inside other clauses, and the way in which lists of adjectives and adverbs are swallowed by nouns and verbs. Context-free grammars are simple enough to allow the construction of efficient parsing algorithms.

4.1 Types of Parsers

A CFG does not specify how to determine whether a given string belongs to the language it defines. To do this, a parser can be used. The main task of a parser is to is to map a string of words to its parse tree [3].

There are mainly two kinds of Parsers, Top- Down Parser and Bottom-up parser.

4.1.1 Top Down Parser

Top- Down parser is a goal oriented parser. Its goal is to parse towards the sentence according to the grammar production. A parse tree is constructed by starting at the start symbol S. A production with S on its left hand side is selected. For each symbol on its right hand side, an appropriate child is constructed. When a terminal is added that does not match the input string, then backtracking is used [1].

Example:

Grammatical Production:

S => NP V

NP => A N | A N O

A ⟶ मोठा | छोटा

N ⟶ हत्ती | राजू

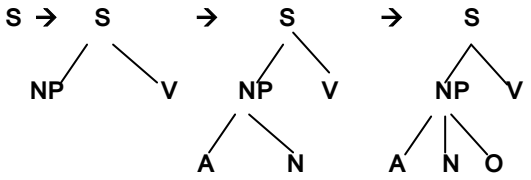
O ⟶ फूटबॉल

V ⟶ आला | खेळतो

Input Sentence :

छोटा राजू फूटबॉल खेळतो

Small Raju plays football.



Decide to use first alternative AN but is wrong

Backtrack and try second alternative

The above example is to show that top down parsing occasionally requires backtracking. The derivation NP- > AN was used at first. Then later, backtracking was done because the derived symbols did not match the input tokens.

4.1.2 Bottom Up Parser

Most Bottom-up parsers are also called as Shift Reduce Parsers. Stack is used and the process of moving word to the stack is called shift and the process of replacing symbol with a nonterminal on the stack is reduce operation.

Bottom Up parsing algorithms start from bottom of the derivation tree and apply grammar rules. Stack is empty at the beginning [3]. The process starts by moving input symbols on to the stack which are then replaced by non-terminals depending on the grammar rules. Once all the symbols are read, the algorithm terminates and the input string is accepted if there is starting nonterminal symbol alone on the stack.

S ⟶ NP V

NP ⟶ A N

A ⟶ मोठा

N ⟶ हत्ती

V ⟶ आला

Table 1. Sequence of Stack in parsing process

$	**मोठा हत्ती आला** **Shift**
$ मोठा	**Reduce** (using A-> मोठा)
$ A	**हत्ती आला** **Shift**
$ A **हत्ती**	**Reduce** (using N -> हत्ती
$ A N	**Reduce** (using NP -> AN)
$ NP	**आला** **Shift**
$ NP **आला**	**Reduce** (using V-> आला)
$ NP V	**Reduce** (using S-> NP V)
$ S	**ACCEPT**

Table 1. shows the sequence of stack frames during bottom up parsing process [1]. Bottom-Up Parser is said to have 2 conflicts such as Shift Reduce and Reduce-Reduce. Top Down parsers are said to be more suitable to parse grammatical productions.

5. CONCLUSION

This paper is an attempt to write context free grammar for simple Marathi sentences. Two sets of examples are taken to explain the writing of CFG. Grammar is parsed with Top Down and Bottom-Up Parser. Top Down parser is said to be more suitable to parse grammatical productions This paper sets a stage to develop computerized grammar checking methods for a given Marathi sentence and stresses mainly on representation of CFG. As future work, representation of Marathi sentences using other types of grammars can be considered.

6. ACKNOWLEDGMENTS

It gives me great pleasure to express my feelings of gratitude to Dr.S.D.Shirbahadurkar, Prof. Ravi Patki and Prof.Sandeep Kadam for valuable support and encouragement. I am thankful to Prof. S.S.Sonone for her suggestions. I owe my sincere feelings of gratitude to Prof.Sandhya Deshpande , HOD, Marathi Dept, for her guidance which helped me a lot to write this paper.

7. REFERENCES

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

[2] Bala Sundara Raman L, Ishwar S, Context Free Grammar for Natural Language constructs - An implementation for Venpa class of Tamil Poetry Tamil Internet 2003, Chennai, Tamilnadu, India

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

[4] Rao, Durgesh, Pushpak Bhattacharya and Radhika Mamidi, "Natural Language Generation for English to Hindi Human-Aided Machine Translation", pp. 179-189, in KBCS-98, NCST, Mumbai.

[5] Gobinda G. Chowdhury, Natural Language Processing Dept. of Computer and Information Sciences University of Strathclyde, Glasgow G1 1XH, UK

[6] Goraksh Garje, Manisha Marathe, Urmila Adsule. *Translation of Simple interrogative sentences to Marathi sentences.*

[7] Dr. Shridhar Shanvare, 'Abhinav Marathi Vyakaran, Marathi Lekhan', Vidya Vikas Mandal, Nagpur

[8] Ayesha Binte Mosaddeque & Nafid Haque, Context-Free Grammar for Bangla, Dhaka, Bangladesh

[9] Bonnie J. Dorr, Pamela W. Jordan, John W. Benoit, „A Survey of Current Paradigms in Machine Translation", LAMP TR-027, Dec. 1998.

[10] R.M.K. Sinha, A.Jain, „Angla Hindi: English to Hindi Machine-Aided Translation System".

[11] Abhijeet R. Joshi, M. Sasikumar, "Constructive approach to teach inflections in Marathi language", www.cdacmumbai.in/design/corporate_site/.../pdf.../CATIML1.pdf