

Bangla Grammar Pattern Detection Using Dynamic Programming Approach

By:

Aroni Saha Prapty

Roll: 1507068

&

MD. Rifat Anwar

Roll: 1507066

A thesis submitted in partial fulfillment of the requirements for the degree of
“Bachelor of Science in Computer Science and Engineering”

Supervisor:

Dr. K.M. Azharul Hasan

Professor

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Signature

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Khulna 9203, Bangladesh

February, 2020

Acknowledgement

The first thank is to the almighty creator who has permitted and blessed us to carry out this study and without whose instruction nothing is possible. The physical and mental fitness also helped us to complete the work which is another supreme favor of God.

We would like to express my deepest sense of gratitude and heartfelt respect and thanks to our honorable teacher and supervisor Prof. Dr. K. M. Azharul Hasan, Department of Computer Science & Engineering, Khulna University of Engineering & Technology who has been a tremendous mentor to us and his continuous guidance, kind advises, sincere motivations and compassionate care has helped us greatly throughout this thesis work.

We thank all the teachers of the Department of Computer Science and Engineering who helped us providing guidelines to perform the work. We would also like to thank our friends and family for their cordial support.

Authors

Abstract

Generating and recognizing the grammar pattern for any natural language is one of the most important and emerging topics in the field of natural language processing. For connecting with civilized world through communication it plays a leading role. Comparing with other techniques, parsing has eminent role in natural language processing. For procreating and recognizing the pattern of Bangla sentence, Context Free Grammar is developed based on the rules of Bangla grammar. This proposed Bangla parser is applied on a specific domain which is Air Traffic Management System implementing dynamic programming parsing algorithm. Most commonly CYK parsing algorithm is used in accordance with adopted parsing grammar for avoiding the problem of ambiguity and left recursion. A syntactical mistake can be recognized by checking parse table if it has no entry for terminal. The correctness of the sentence is identified if it is successfully parsed and reached the start symbol based on the grammatical rules of Bangla language.

Contents

	Page
Title Page	i
Acknowledgement	ii
Abstract	iii
Contents	iv
List of Tables	vi
List of Figures	vii
CHAPTER 1 Introduction	1
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Objectives	2
1.4 Scope	3
1.5 Organization of Thesis	3
CHAPTER 2 Literature Review	4
2.1 Introduction	4
2.2 What Is NLP	4
2.2.1 Applications of NLP	5
2.2.2 Steps of NLP	5
2.3 Background	6
2.3.1 Context Free Grammar	6
2.3.2 Chomsky Normal Form	7
2.3.3 CYK Algorithm	8
2.4 Related Works	11
2.5 Discussion	11
CHAPTER 3 CYK Based Parsing on Specific Domain	12
3.1 Introduction	12
3.2 Development of Context Free Grammar	13

3.3	Building CNF from CFG	17
3.3.1	Removing Start Symbol from Right Hand Side	17
3.3.2	Removing Null Productions	18
3.3.3	Removing Terminal with Non-Terminal or Terminal	19
3.3.4	Removing Two or More Non-Terminals from R.H.S	19
3.4	Dynamic Parsing: CYK Parsing Algorithm	20
3.4.1	Dynamic Parsing of Declarative Sentence	21
3.4.2	Dynamic Parsing of Imperative Sentence	23
3.4.3	Dynamic Parsing of WH Question	24
3.4.3	Dynamic Parsing of Yes-No Answer Type Question	27
3.5	Pseudocode	29
CHAPTER 4	CYK Based Parsing on Specific Domain	30
4.1	Introduction	30
4.2	Data Collection	30
4.3	Result Analysis	38
4.4	Comparison between Dynamic Parsing and Other Parsing	43
CHAPTER 5	Conclusion	44
5.1	Summary	44
5.2	Limitations	44
5.2	Future Works	44
References		46
Appendix		48

List of Tables

Table No.	Description	Page
3.1	Noun Phrase Identification	14
3.2	Verb Phrase Identification	15
3.3	POS Tag Description	15
3.4	Complete Production Rule	16
4.1	Input Data Table	30
4.1.1	Input Data Table (Declarative Sentences)	30
4.1.2	Input Data Table (Yes-No-Question Sentences)	33
4.1.3	Input Data Table (WH-Question Sentences)	35
4.1.4	Input Data Table (Imperative Sentences)	37
4.2	Performance Measurement	42

List of Figures

Figure No.	Description	Page
2.1	Left Most Derivation	6
2.2	Right Most Derivation	6
2.3	Bottom Row Construction	10
2.4	2nd Row Construction	10
2.5	Top Row Construction	10
3.1	Parse Tree for Bangla Sentence	13
3.2	Bottom Row Construction	22
3.3	2nd Row Construction	22
3.4	2nd Row Construction	23
3.5	Top Row Construction	23
3.6	Bottom Row Construction	23
3.7	Top Row Construction	23
3.8	Bottom Row Construction	25

3.9	2nd Row Construction	25
3.10	Top Row Construction	25
3.11	Bottom Row Construction	27
3.12	2nd Row Construction	27
3.13	Top Row Construction	27
4.1	Output for Declarative Sentence	41
4.2	Output for Imperative Sentence	41
4.3	Output for Yes-No Answer Type question	42
4.4	Output for WH question	42
4.5	Output for Improper Input	42

CHAPTER 1

Introduction

1.1 Introduction

To express the feelings and communicate with others, language is an effective system of sounds, gestures, signs and symbols. Communicating through a language one should understand the language, know about the proper structure of the language and use the correct grammatical form of that language for appropriate meaning. Bangla is one of most powerful Indo-Iranian languages that is internationally recognized and spoken by more than 250 million [9]. But in the field of Natural Language Processing it has not given the priority for using in computational linguistics. Grammar checking is important for any kind of language to express the sentiments and retrieve information. For this it should be syntactically and semantically correct.

For checking syntax correctly parsing is one of the best methods. Parsing is an activity of breaking down sentences into parts to check the grammatical correctness of a given sentence using specific algorithm for the given context free grammar. Predictive parsing is one of the most used parsing algorithms [1] [2] that has some issues on ambiguity and left factoring when the given grammar is left recursive. Shift reduce parsing has overcome those issues but still conflicts when a state requests for both shift and reduce actions or more than one distinct reduce actions [11] which can be handled by left factoring for top down parsing. For avoiding left factoring and ambiguity issue and checking correctness of Bangla sentence we apply basic idea of the dynamic parser CYK algorithm where parse tree follows the path from bottom to top according to the context free grammar and if it can reach the start symbol it is considered that the sentence is grammatically correct. This process is challenging for Bangla language because one sentence which has same semantic meaning can be represented in different patterns as there is no fixed

structure and even same word can be represented as different way [3] and for this the meaning of the sentence will change. CFG for Bangla language is huge hence we develop CFGs for affixed domain. We select Air traffic management system (ATMS) as our domain to check the grammar pattern. The scheme is applicable to recognize grammar [1], text conversion [4], machine translation [7], syntax analysis [8], questions answering and so on.

1.2 Problem Statement

Bangla Grammar pattern detection is nothing but find the structure a sentence following the context free grammar rules. As Bangla language is huge in form so we take a specific domain that is Air Traffic Management System (ATMS) and develop a CFG (context free grammar) for this grammar. There are many approaches for find the correctness of the Bangla sentence. Such as predictive parsing, shift reduce parsing, unsupervised morphological parsing but each has some issue for ambiguity and left factoring if the grammar is left recursive. To overcome this issue a dynamic programming algorithm named CYK (Cocke – Younger - Kasami) parsing algorithm is used to check the correctness of Bangla sentence. The motive of this work is to make machine to understand the natural language

1.3 Objective:

Bangla Grammar Recognition is a part of Natural Language Processing and language processing is huge field of research to make the machine capable of understanding the human spoken languages. The process of understanding the natural language is a very difficult job for machine. In this work we have worked with Bangla language processing which is very immature research area. Many have tried with English or other language processing but works with Bangla language processing is very rare. Our research work has several objectives to be full filled and several of them have been mentioned in the following section:

- Input a sentence in Bangla language and determine whether the sentence is grammatically right or wrong.
- Tokenizing the input sentence and categorizing the tokens into Bangla Parts of Speech.
- Implement with CYK algorithm for matching purpose.

- Prepare machine to produce an effective result on the basis of Natural Language Understanding.

1.4 Scope

In our thesis we have worked for Bangla language to check whether the input sentence is grammatically correct or wrong. As natural language processing is a branch of artificial intelligence and the interaction between human and computer through natural language. To understand the natural language it should be syntactically and semantically right. The scopes of our thesis are:

- **Selected Domain:** ‘Air Traffic Management System’ shorted as ‘ATMS’.Only considers conversation about local Air services.
- **Selected Language:** For Bangla Language.
- **Selected IDE for Development:** Python 3.7. It is very easy for handling Bangla Language.
- **Selected Software:** We used Pycharm Community Edition 2018.3.4 x64 to run Python IDE

1.5 Organization of Thesis

The remaining part of this thesis consists of the parts as follows:

Chapter 2: It represents literature review that describes some of the related works on Bangla grammar pattern recognition.

Chapter 3: Mechanism of syntactic analysis on Bangla language is discussed in this chapter.

Chapter 4: The experimental outcomes and performance evaluation of our method are explained in this chapter.

Chapter 5: The future scope of this work and the conclusive words about the method are outlined in this chapter.

Chapter 2

Literature Review

2.1 Introduction

In recent decades, natural language processing has great impact on various research fields. So study on this topic has started from the very beginning. But very few research works are done in Bangla language. So our main aim is to perform on Bangla language and check whether an input sentence is grammatically right or wrong. For this a context free grammar is developed for a domain named air traffic management system (ATMS) using CYK parsing algorithm.

2.2 What Is NLP

NLP is the combination of Artificial Intelligence, machine learning and linguistics that allows us to interact to machines as if they were human.

In this 21st century industrial estimates that only 20% data are presented in a structured form. Data is generated in form of speak, images, messages and in many more forms. Majority of data is accessed in textual form which is highly unstructured in nature. In order to produce significant and actionable information from data, it is necessary to acquire knowledge about text analysis and natural language processing. Text mining is the process of deriving information from natural language. It usually involves the process of structuring input text deriving patterns in structured data and finally evaluates an interpreting output. On the other hand natural language processing refers an artificial intelligence method of communicating with intelligent system using human languages. Nowadays there are many fields of real life where NLP can be implemented for data mining and information gathering.

2.2.1 Applications of NLP

- Machine translation
- Sentiment analysis
- Speech recognition
- Spell checking
- Keyword searching
- Information extraction
- Implementation of Chabot
- Question answering

2.2.2 Steps of NLP

NLP is divided into two major components:

- Natural Language Understanding
- Natural Language Generation

Natural language Understanding means the mapping of given input into natural language for useful representation and analyzing those aspects of the language. On the other hand Natural Language Generation is a process of producing meaningful phrases and sentences in the form of natural language for some internal representation .Natural Language Understanding is harder than natural Language Generation because it takes lot of times and things in a particular language specially for intelligent devices. Steps of Natural Language Processing are given below:

- Tokenization
- Stemming
- Lemmatization
- POS Tag
- Name Entity Recognition
- Chunking

2.3 Background

2.3.1 Context Free Grammar

To generate the pattern of strings context free grammar is used and it is a set of recursive rules. To describe context free language, compiler design and programming language context free grammar is used. Context free language is constrained by specific rules and it is a set of strings of symbol which is generated by context free grammar. Context-free-grammar consists of the following quadruple (N, T, R, S):

- S: S is a start and special non-terminal symbol of the production.
- N: It is a set of non-terminal symbols.
- T: It is a set of terminal symbols.
- R: It is a set of production rules, where non-terminal symbols of the left side of the production are replaced by the non-terminal symbol or terminal symbol of the right side of the production.

Let consider the following rules of grammar:

$S \rightarrow S * S \mid S + S \mid x \mid y \mid z$

Here, S is a start symbol which is non-terminal and x, y, z are the set of terminal symbols.

Derivation Tree of $x + y * z$

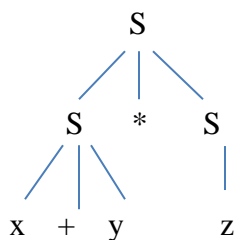


Figure 2.1: Left Most Derivation

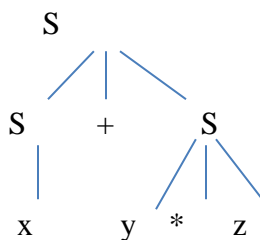


Figure 2.2: Right Most Derivation

2.3.2 Chomsky Normal Form

A context free grammar is considered as CNF if the left hand side of the production rule consists of either two non-terminal symbols or one terminal symbol or both.

Suppose: $S \rightarrow AB / a$

This production rule is in the form of CNF. If the grammar is not in the CNF form then the following rules are applied to convert the grammar in CNF form:

- If the start symbol is S then new start symbol is S_0 , so the new production rule is $S_0 \rightarrow S$.

- Then Null production should be eliminated.

$S_0 \rightarrow S$

$S \rightarrow BAS$

$B \rightarrow b / \epsilon$

$A \rightarrow B / aC$

Put ϵ to the right side of the production of S . Then the following production is:

$S_0 \rightarrow S$

$S \rightarrow BAS / AS / BA / S$

$B \rightarrow b$

$A \rightarrow B / aC$

- If there are any unit productions then it should be eliminated. Such that:

$S \rightarrow S$ and $A \rightarrow B$ are unit productions and they replaced by the right hand side of the production S and B respectively.

$S_0 \rightarrow BAS / AS / BA$

$S \rightarrow BAS / AS / BA$

$B \rightarrow b$

$A \rightarrow b / aC$

- If there are more than two non-terminals in the right side of the production then it should be replaced as the form of CNF. Such that:

$S \rightarrow BAS$ should be replaced in the following way:

$So \rightarrow PA / AS / BA$

$S \rightarrow PA / AS / BA$

$B \rightarrow b$

$A \rightarrow b / aC$

$P \rightarrow BS$

- If the right side of the production contains one terminal and one non-terminal symbol then terminal symbol is replaced by a non-terminal symbol.

$So \rightarrow PA / AS / BA$

$S \rightarrow PA / AS / BA$

$B \rightarrow b; \quad A \rightarrow b / XC$

$P \rightarrow BS; \quad X \rightarrow a$

2.3.3 CYK Algorithm

There are two types of parsing. These are:

- i. Top-down parsing: Parsing from top to bottom also known as goal driven parsing.
- ii. Bottom-up parsing: Parsing from bottom to top also known as data-driven parsing.

Naive VS Dynamic Programming:

- Naive: It enumerates everything.
- Backtracking: If there are partial solutions it discards it.
- Dynamic Programming: If there are partial solutions it saves it in the table.

Configuring the parsing techniques CYK is bottom up parsing using dynamic programming approach.

CKY algorithm is known as the parsing and recognition algorithm. Recognition is a process that decides if there are any memberships in a particular language. Parsing is a combination of recognition which analyzes the input to generate a data according to the production of the grammar and produces a parse tree. CKY algorithm is a bottom up parsing algorithm that means it generates the parse tree starting from the leave nodes. This algorithm is mostly used for

recognition (existence of a parse tree) and probabilistic parsing (most probable parse tree). CKY algorithm is applicable if the CNG is in the form of CNF.

CYK parsing algorithm is recognized when the language is defined by the context free grammar and is in the form of CNF. It is harder to handle if it is not in the form of CNF. So it uses dynamic programming approach compositionally to build solution from sub solutions.

- We have to consider all possible consecutive sequence of production rules that is made of letters and sets $R \in P[i, j]$ if the production rule can be obtained from non-terminal R.
- It sequentially go through the length will start from one upto the total words in the sentence.
- If the length is two or greater than two it creates the partition of words according to the number of rows and checks if it matches any production rule.
- When the process is completed and if the string generates the start symbol it is considered that the sentence is recognized by the grammar.

Consider the following CNF form of the grammar G.

$S \rightarrow AA$

$A \rightarrow a / AB$

$B \rightarrow b$

$C \rightarrow AB / AC$

Is string “aab” belongs to the grammar?

Construction of Triangular Table:

$$Z_{m,n} = (Z_{m,m}, Z_{m+1,n}), (Z_{m,m+1}, Z_{m+2,n}) \dots (Z_{m,n-1}, Z_{n,n}) \dots \dots \dots (1)$$

The length of the bottom row is one, second from the bottom row length is two and top row the length is the total word in the input sentence (string ‘w’).

$Z_{m,n}$ is the set of variables K such that $P \rightarrow w_m$ is the production rule of grammar G. So, the previously computed n pair sets:

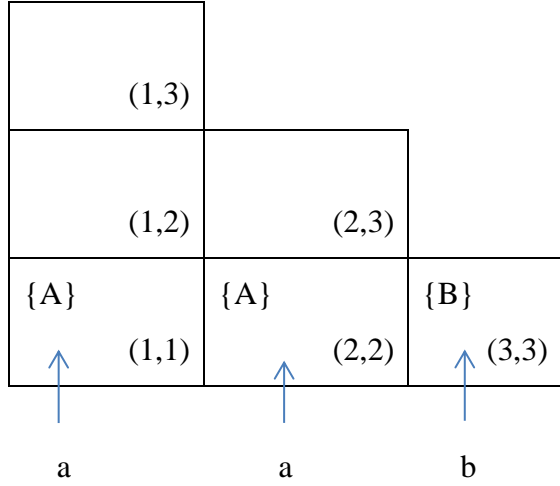


Figure 2.3: Bottom Row Construction

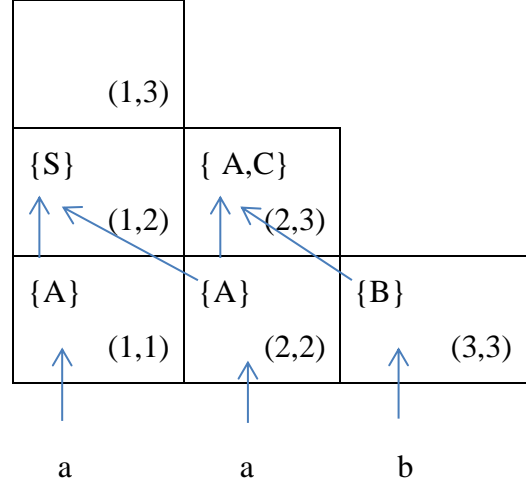


Figure 2.4: 2nd Row Construction

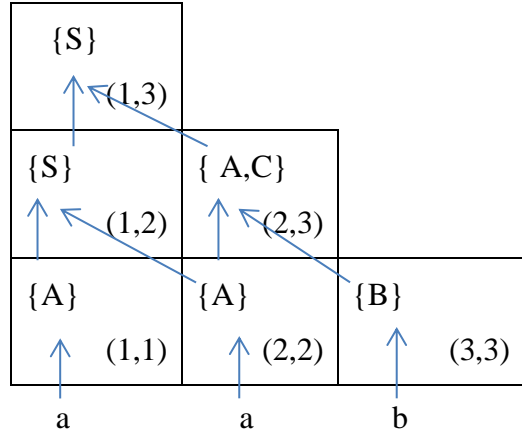


Figure 2.5: Top Row Construction

Figure- 2.4 is constructed by using context free grammar G. The construction process of Figure: 2.5 using equation 1 is given below:

$$Z_{1,2} = (X_{m,m}, X_{m+1,n}) = (X_{1,1}, X_{2,2}) \rightarrow \{A\}\{A\} = \{AA\} = \{S\}$$

$$Z_{2,3} = (X_{m,m}, X_{m+1,n}) = (X_{2,2}, X_{3,3}) \rightarrow \{A\}\{B\} = \{A,C\}$$

There are two production rules that construct $\{AB\}$. So, $Z_{2,3} = \{A, C\}$;

The construction process of Figure: 2.6 using equation 1 is given below:

$$\begin{aligned}
Z_{1,3} &= (Z_{m,m}, Z_{m+1,n}) (Z_{m,m+1}, Z_{m+2,n}) \\
&= (Z_{1,1}, Z_{2,3}), (Z_{1,2}, Z_{3,3}) \\
&= \{A\}\{A,C\} \cup \{S\}\{B\} \\
&= \{AA,AC,SB\} \\
&= \{S\}
\end{aligned}$$

From the calculation it reaches to the start symbol so the string “aab” belongs to the following grammar G.

2.4 Related Works

For parsing Bangla sentence predictive parser is applied in [1][2] that cannot work for ambiguous grammar. To check the rule probability and word probability of a Bangla sentence CYK parsing algorithm is used in [3] to fit it the most probable structure of the sentence. For mapping and translating Bangla language with other natural languages predicate preserving parser is discussed in [4]. Using the basic of linguistic knowledge builder the Head Driven Phase Structure Grammar is described in [5] that can handle semantics of Bangla sentence. For parsing a Bangla sentence an unsupervised morphological technique is proposed in [6]. Another Bangla morphological analysis is done in [7] for final state technology. Syntactical analysis of a context sensitive Bangla text is done in [8] for machine translation using NLP conversion toolkit. LR parsing is used in [9] for parsing Bangla grammar which is time consuming and cannot handle all left recursive grammar. For avoiding ambiguity and left factoring shift reduced parser is used in [10] but conflicts for a state that requests more than one specific reduce actions or both shift and reduce actions.

2.5 Discussion

After analyzing the parsing techniques we see that CYK parsing algorithm gives the better result from other parsing because it can deal with the ambiguity issue, left factoring if the grammar is left recursive and it is not time consuming at all. So we implement the CYK parsing algorithm for getting the better performance of our work.

Chapter 3

CYK Based Parsing on Specific Domain

3.1 Introduction

Our purpose is to detect the correctness of any natural language and for this parsing is one the best techniques. Parse tree is a tree that graphically represents the derivation of the grammar of the input string and the process is called parsing. We have worked with Bangla language to check the appropriateness of a sentence using a bottom-up parsing algorithm for a specific domain named “Air Traffic Management system”. In our experiment the proposed algorithm is based on idea of **Cocke-Younger-Kasami** parsing algorithm. The **Cocke-Younger-Kasami** algorithm (also called CYK or CKY) is a bottom-up parsing algorithm for **Context-Free-Grammars** (also called CFGs) which follows a dynamic approach for parsing. An input sentence is given to our system and the system generates a parse tree according to the context free grammar for this specific domain. The validity of the sentence is determined if the parse tree generates start symbol.

The reason of choosing CYK parsing algorithm is as follows:

- The parsing is not affected by the ambiguity of CFG grammar.
- It can handle left recursive (LR) grammars.
- The CFG does not need to be left factored.

To implement CYK parsing algorithm for Bangla language, we need CFG rules.

3.2 Development of Context Free Grammar

Context-Free-Grammar or Phrase-structure Grammar is extensively used for modeling the formation of any natural language. It comprises set of production or rules, each of which explicits the form that tokens of the language can be systematized. For the following sentence a parse tree is generated according to CFG.

আমি ঢাকার বিমানের টিকিট চাই

$S \rightarrow NP VP;$

$NP \rightarrow Pronoun \mid Noun \mid Pronoun Adjective Noun;$

$VP \rightarrow TV;$

$Adjective \rightarrow Adjective Adj \mid Adj;$

$Pronoun \rightarrow$ আমি;

$Noun \rightarrow$ টিকিট;

$Adj \rightarrow$ বিমানের/ ঢাকার;

$TV \rightarrow$ চাই

The formal grammar that is used in natural language processing is known as generative grammar. Here the formal grammatical rules of Bangla language is used where each terminal symbol has POS tags itself and goal is reach the start symbol.

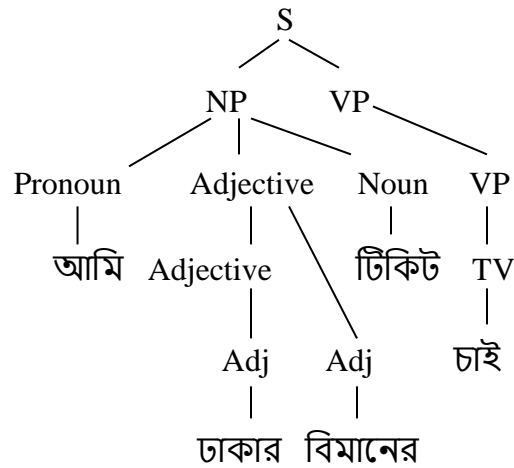


Figure 3.1: A parse Tree for “আমি ঢাকার বিমানের টিকিট চাই”

For convenient and compact format representation of parse tree bracketed notation is used. For the given example bracketed notation is: [S[NP[Pronoun আমি][Adjective[Adjective[Adj ঢাকার]][Adj বিমানের]]][Noun টিকিট]][VP[TV চাই]]]

Any sentence in the formal language that can be obtained from production rules defined by CFG is referred as **grammatical** sentence. Besides sentences that cannot be derived by CFG is acknowledges as **ungrammatical** sentences. It is difficult to explain the whole structure of a natural language (say, Bangla) but a simplified model can be generated based on the context. By utilizing formal language to build up natural languages is defined as **generative grammar** as the sentences of the language is generated by CFG. The proposed grammar for ATMS domain contains non-terminals like NP (noun phrase), VP (verb phrase), AP (adjective phrase). The terminal symbol can be derived from the non-terminal symbols. Noun phrase contains noun or pronoun as a head word. Noun phrase can be single noun or accompanied by determiners, modifiers or some complements. Adjective phrase contains one more adjectives and can be accompanied by modifiers, determiners or other words. The verb phrase contains one or more words beside the verb. Verb can be transitive verb or intransitive verb. The following table-I, table-II, table-III contain some specimens of noun phrase and verb phrase and the POS tag description with example.

Table 3.1: Noun Phrase Identification

SL. No	Noun Phrase	Identifying Noun
1.	ঢাকার টিকেট	টিকেট
2.	ঢাকা থেকে খুলনা	ঢাকা, খুলনা
3.	যাতায়াতের মালামাল	মালামাল

Table 3.2: Verb Phrase Identification

SL. No	Verb Phrase	Identifying Verb
1.	টিকিট চাই	চাই
2.	পাসপোর্ট দেখান	দেখান
3.	বিমান দেরিতে ছাড়বে	ছাড়বে

Table 3.3: POS Tag Description

SL. No	Parts-Of-Speech	Symbol	Example
1.	Noun	Noun	বিমান, মালামাল
2.	Pronoun	Pronoun	আমি, আপনি
3.	Adjective	Adjective	ভালো, অতি আরামদায়ক
4.	Transitive, Intransitive Verb	TV, IV	যাবো, খাবো (TV) যেতে, দিতে (IV)
5.	Adverb	Adv	দ্রুত, ধীরে
6.	Conjunction	Conj	এবং, ও
7.	Preposition	Pre	থেকে, হতে
8.	Modifiers	Modifiers	একটি
9.	Negative Word	Neg	না, নেই
10.	WH Question	Wh	কি, কখন, কবে
11.	Yes No Question	Aux	কি

This POS tag symbol is used in table-IV where all the production rules are combined for parsing the Bangla sentences for the fixed ATMS domain.

Table 3.4: Complete Production Rules

S -> NP VP| NP VP Aux| NP Wh VP| WH NP VP| Adverb VP| NP WH

NP -> Pronoun Noun| Det Noun| Pronoun NP| Pronoun Aux| Adjective Noun| Adjective NP| NP
NP| Noun Noun| Noun| NP Noun| Noun Aux| NP Aux| Noun Aux Noun| NP Aux Noun| Pronoun
NP| NP Adjective| Pronoun| Pronoun Aux Pronoun| Pronoun Aux Adjective Noun| Pronoun Aux
NP| Pronoun Wh| Wh Noun| Pronoun Wh Noun| Wh Noun Noun| Pronoun Wh NP| Wh Noun
Pronoun| Pronoun Adjective Noun| Modifier Noun| Pronoun Modifier Noun| Noun Adjective
Noun| Noun Modifier Noun| Pronoun Modifier Noun| Noun NP| Noun Aux NP| Noun Aux
Adjective Noun| Adjective Det Noun| Pronoun| Aux Pronoun| Pronoun Aux Pronoun
Noun| Pronoun Noun Noun| Pronoun Noun Wh| NP Wh| Pronoun Noun Wh| Noun Conj
Noun| Pronoun Conj Noun| Pronoun Conj Pronoun| Pronoun Noun Conj Noun| Adjective Noun
Wh| Adjective Noun Conj| Pronoun Aux Pronoun Noun| Pronoun Pronoun| Pronoun Pronoun
Noun| Pronoun Noun Adjective Noun| Pronoun Aux NP| Pronoun Aux Noun| Adjective Noun
Adjective Noun| Noun Adjective Noun Conj| Noun Pronoun Conj Pronoun| Noun Pronoun
Adjective| Noun Noun Conj Pronoun| Noun Noun Conj Pronoun| Pronoun Adjective| Pronoun
Pronoun conj Pronoun| Modifier Conj Adjective| Noun Pre Noun| Noun Pre| AP Noun| Pronoun
Aux Adverb| Adverb Noun| Noun Adverb

VP -> IV TV| Noun IV| TV| Adverb TV| TV Aux| Adjective Noun TV| Noun TV| Noun Noun
TV| Adverb IV| IV IV TV| Noun Aux Adverb IV| Adverb IV IV TV| Adverb IV IV| Adjective
Noun Aux Adverb IV| Noun Aux Adverb IV IV| Noun Aux Adverb IV| Pronoun IV| Wh Noun
Pronoun IV| Noun TV| Adjective Noun TV| Modifier Noun TV| Adjective Noun IV| Noun VP|
Noun IV TV| Noun Adjective Noun IV| TV Aux| VP Aux| IV TV Aux| Adverb TV Aux| Noun
Adjective Noun IV| Wh TV| Adverb TV| Noun Conj Noun IV| Pronoun Aux VP| Noun IV TV
Neg| Noun TV Neg| TV Neg| Noun TV Adverb| IV TV Adverb| Pronoun Adjective TV| Pronoun
IV TV Neg| Noun TV TV| Pronoun TV TV| Pronoun TV TV Neg| TV Adjective Noun| IV TV
Adjective Noun| TV Adjective Pronoun| IV TV Adjective Pronoun| Adjective Noun TV
Neg| Adjective TV| Adverb IV TV| Modifier Noun IV| Det Noun IV| IV TV Neg

AP -> Adjective Noun| Adjective Pronoun| Adjective Neg| AP Adjective

CYK parsing cannot work with CFG rules. To work with CYK parsing, we need to convert CFG rules to **Chomsky-Normal-Form**.

3.3 Building CNF from CFG

Chomsky-Normal-Form (also called CNF) can be seen as a subset of Context-Free-Grammar. A CFG can be said to be in CNF if it satisfies one of the conditions mentioned below:

- ✓ A Non-Terminal symbol will generate a Terminal symbol (e.g.; $A \rightarrow a$;))
- ✓ A Non-Terminal symbol can generate two Non-Terminal symbols (e.g.; $A \rightarrow DF$;))
- ✓ The symbol designated as the Start symbol will generate ϵ (e.g.; $\text{Start} \rightarrow \epsilon$;))

Now consider a CFG rule in ATMS in Bangla.

The CFG rule is as follows:

$S \rightarrow NP VP / NP VP S$;

$NP \rightarrow \text{Pronoun Noun}$;

$VP \rightarrow IV IV TV$;

$IV \rightarrow \epsilon \mid \text{পেতে}$;

$TV \rightarrow \text{চাই}$;

$\text{Pronoun} \rightarrow \text{Pronoun} \text{ আমি}$;

$\text{Noun} \rightarrow \text{টিকিট}$;

Now with the help of it, the steps of the transformation of CFG to CNF will be discussed.

3.3.1 Removing Start Symbol from Right Hand Side

If the start symbol (Here; S) can be found at the right-hand side of any of the production rules, then a new production rule will be created as:

$S_0 \rightarrow S$; # Here the start symbol is changed from S to S_0 .

After adding the rule, the resulting grammar rule will look like the following:

$S_0 \rightarrow S$;

$S \rightarrow NP VP / NP VP S$;

$NP \rightarrow \text{Pronoun Noun}$;

$VP \rightarrow IV IV TV;$

$IV \rightarrow \epsilon \mid \text{পেতে};$

$TV \rightarrow \text{চাই};$

$Pronoun \rightarrow Pronoun \text{আমি};$

$Noun \rightarrow \text{টিকিট};$

3.3.2 Removing Null Productions

Observing the production rules, we see the rule $IV \rightarrow \epsilon$. After removing this, the grammar rule becomes:

$S_0 \rightarrow S;$

$S \rightarrow NP VP \mid NP VP S;$

$NP \rightarrow Pronoun Noun;$

$VP \rightarrow IV IV TV \mid IV TV \mid TV;$

$IV \rightarrow \text{পেতে};$

$TV \rightarrow \text{চাই};$

$Pronoun \rightarrow Pronoun \text{আমি};$

$Noun \rightarrow \text{টিকিট};$

Now we have two unit production rules $S_0 \rightarrow S$; and $VP \rightarrow TV$; Remove unit product S and TV with their production rule yield the given grammar as:

$S_0 \rightarrow NP VP \mid NP VP S;$

$S \rightarrow NP VP \mid NP VP S;$

$NP \rightarrow Pronoun Noun;$

$VP \rightarrow IV IV TV \mid IV TV \mid \text{চাই};$

$IV \rightarrow \text{পেতে};$

$TV \rightarrow \text{চাই};$

$Pronoun \rightarrow Pronoun \text{আমি};$

$Noun \rightarrow \text{টিকিট};$

3.3.3 Removing Terminal Symbols Attached with Non-Terminal or Terminal symbols

The production rule '*Pronoun* \rightarrow *Pronoun* আমি' is found in the CFG rule we have taken as an example. Here, terminal symbol 'আমি' is with a Non-Terminal symbol '*Pronoun*'. This rule can be festered as:

Pronoun \rightarrow *Pronoun* X_0 ;

$X_0 \rightarrow$ আমি;

Now, our grammar rule looks as follows:

$S_0 \rightarrow NP VP$;

$S \rightarrow NP VP$;

$NP \rightarrow Pronoun Noun$;

$VP \rightarrow IV IV TV / IV TV / চাই$;

$IV \rightarrow পেতে$;

$TV \rightarrow চাই$;

Pronoun \rightarrow *Pronoun* X_0 ;

$X_0 \rightarrow$ আমি;

Noun \rightarrow টিকিট;

3.3.4 Remove Two or More Non-Terminals from RHS

If a production rule exists like (e.g.; $A \rightarrow BCD$); then it can be decomposed as:

$A \rightarrow X_1 D$;

$X_1 \rightarrow BC$;

The example CFG we are using for our ATMS, there are three production rules like (e.g.; ' $A \rightarrow BCD$ ') are ' $S_0 \rightarrow NP VP S$ ', ' $S \rightarrow NP VP S$ ' and ' $VP \rightarrow IV IV TV$ '. These rules can be decomposed as:

$S_0 \rightarrow X_1 S$;

$X_1 \rightarrow NP VP$;

$S \rightarrow X_1 S$;

$VP \rightarrow X_2 TV$;

$X_2 \rightarrow IV IV$;

So, the final result of our example grammar rule in the CNF form is:

$S_0 \rightarrow NP VP / X_1 S;$

$S \rightarrow NP VP / X_1 S;$

$X_1 \rightarrow NP VP;$

$NP \rightarrow \text{Pronoun Noun};$

$VP \rightarrow X_2 TV / IV TV / \text{চাই};$

$X_2 \rightarrow IV IV;$

$IV \rightarrow \text{পেতে};$

$TV \rightarrow \text{চাই};$

$\text{Pronoun} \rightarrow \text{Pronoun } X_0;$

$X_0 \rightarrow \text{আমি};$

$\text{Noun} \rightarrow \text{টিকিট};$

This is the final CNF form from our example CFG rules.

3.4 Dynamic Parsing: CYK Parsing Algorithm

For implementing the Dynamic-Parsing, context free grammar (CFG) is generated for the specific domain and CFG is converted to Chomsky normal form (CNF). Then a triangular table is made of using CNF according to the idea of CYK parsing algorithm. The size of the parse tree depends on the number of terminal symbols in the input sentence. During parsing terminal symbol is replaced by the left side of the produced grammar and the number of replacements of the terminal symbol depends on the number is row. After passing through the entire input for each row one column is reduced from the last and at the very last if the terminal symbols can be replaced by the start symbol; then it is considered that the given input is valid. We will discuss this idea by four types of sentences; These are:

- ✓ Declarative Sentence (Grammar rules like: $S \rightarrow NP VP;$)
- ✓ Imperative Sentence (Grammar rules like: $S \rightarrow NP VP;$)
- ✓ Yes-No Questions (Grammar rules like: $S \rightarrow NP Aux VP / NP VP Aux;$)
- ✓ WH Questions (Grammar rules like: $S \rightarrow NP WH VP / NP VP WH;$)

3.4.1 Dynamic Parsing of Declarative Sentence

To illustrate the process of dynamic parsing of declarative sentence first we take an example in ATMS in Bangla language. That is:

‘আমি ঢাকা যাব’

The CNF form of the rule for parsing the sentence is:

$S \rightarrow NP VP;$

$NP \rightarrow \text{Pronoun Noun} / \text{Noun} / \text{Pronoun};$

$VP \rightarrow TV;$

$\text{Pronoun} \rightarrow \text{আমি};$

$\text{Noun} \rightarrow \text{ঢাকা};$

$TV \rightarrow \text{যাব};$

The bottom row is constructed using the grammar G in figure-3.2.

From the equation (1) we found:

$$Z_{m,n} = (Z_{m,m} Z_{m+1,n}), (Z_{m,m+1} Z_{m+2,n}) \dots (Z_{m,n-1}, Z_{n,n})$$

$$Z_{1,2} = (Z_{m,m}, Z_{m+1,n}) = (Z_{1,1}, Z_{2,2})$$

$$\begin{aligned} &= \{NP, \text{Pronoun}\} \{NP, \text{Noun}\} \\ &= \{NP NP, NP \text{Noun}, \text{Pronoun NP}, \text{Pronoun Noun}\} \\ &= \{NP\} \end{aligned}$$

Steps:

- Looking for rules of productions that generate ‘NP NP, NP Noun, Pronoun NP, Pronoun Noun.’
- There is one production, that is ‘NP \rightarrow Pronoun Noun’.
- $Z_{1,2} = \{NP\}$

$$Z_{2,3} = (Z_{m,m}, Z_{m+1,n})$$

$$\begin{aligned} &= (Z_{2,2}, Z_{3,3}) \\ &= \{NP, \text{Noun}\} \{VP, TV\} \\ &= \{NP VP, NP TV, \text{Noun VP}, \text{Noun TV}\} = \{S\} \end{aligned}$$

Steps:

- Looking for rules of productions that generate ‘NP VP, NP TV, Noun VP, Noun TV’.
- There are one productions, that is ‘S -> NP VP’.
- $Z_{2,3} = \{S\}$

The construction of $Z_{1,2}$ and $Z_{2,3}$ is shown respectively in figure-3.3 and figure-3.4.

$$\begin{aligned}
 Z_{1,3} &= (Z_{m,m}, Z_{m+1,n}) (Z_{m,m+1}, Z_{m+2,n}) \\
 &= (Z_{1,1}, Z_{2,3}), (Z_{1,2}, Z_{3,3}) \\
 &= \{NP, Pronoun\} \{S\} \cup \{NP\} \{VP, TV\} \\
 &= \{NP S, Pronoun S, NP VP, NP TV\} \\
 &= \{S\}
 \end{aligned}$$

Steps:

- Looking for rules of productions that generate ‘NP S, Pronoun S, NP VP, NP TV.’
- There is one production, that is ‘S -> NP VP’.
- $Z_{1,3} = \{S\}$

The construction of $Z_{1,3}$ is shown in figure-3.5.

As Start symbol ‘S’ is generated so the given input sentence is grammatically correct.

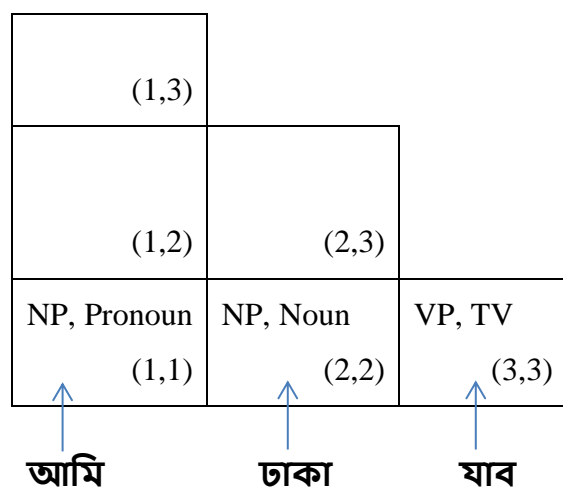


Figure 3.2: Bottom Row Construction

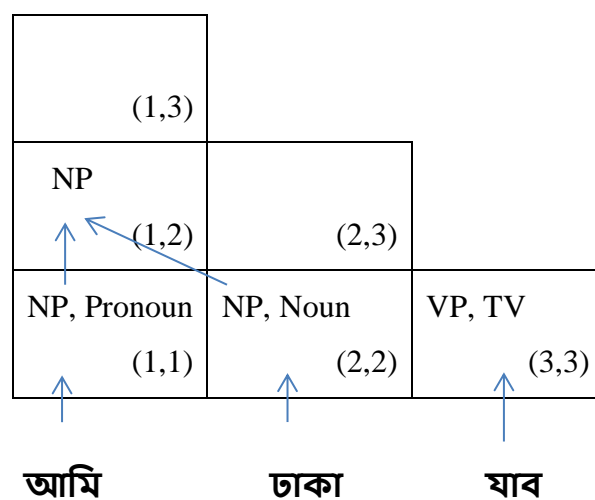


Figure 3.3: 2nd Row Construction

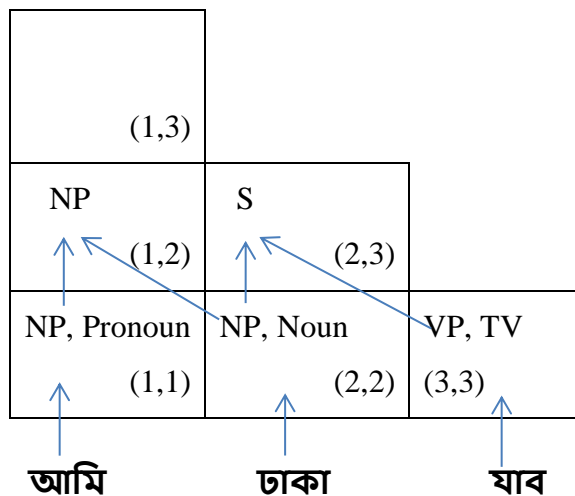


Figure 3.4: 2nd Row Construction

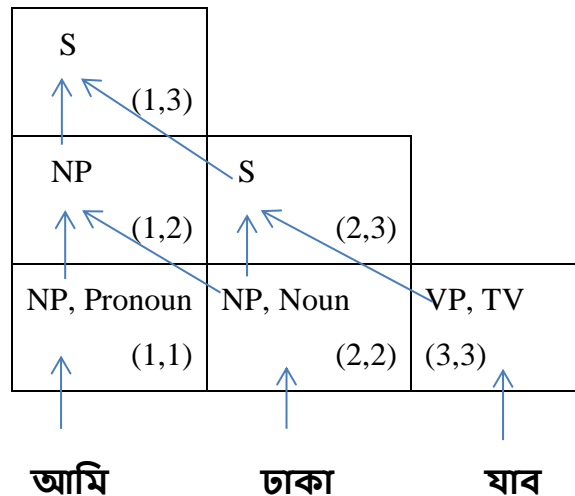


Figure 3.5: Top Row Construction

3.4.2 Dynamic-Parsing of Imperative Sentence

To illustrate the process of dynamic-parsing of imperative sentence again we take an example in ATMS in Bangla language. That is:

‘টিকিট চাই’

The context free grammar (CFG) of the sentence is:

$S \rightarrow NP VP;$

$NP \rightarrow Noun;$

$VP \rightarrow TV;$

$Noun \rightarrow$ টিকিট;

$TV \rightarrow$ চাই;

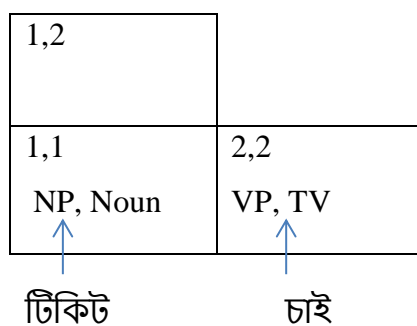


Figure 3.6: Bottom Row Construction

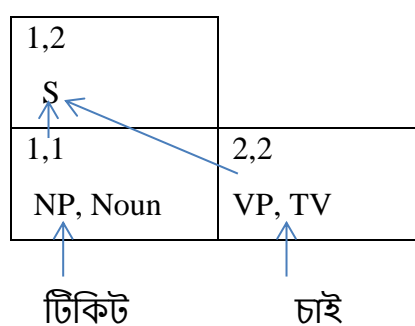


Figure 3.7: Top Row Construction

The construction of $Z_{1,1}$, $Z_{1,2}$ are done by using CFG and shown in figure-3.6. From the equation (1) we found;

$$Z_{m,n} = (Z_{m,m} Z_{m+1,n}), (Z_{m,m+1} Z_{m+2,n}) \dots (Z_{m,n-1}, Z_{n,n})$$

$$\begin{aligned} Z_{1,2} &= (Z_{m,m}, Z_{m+1,n}) = (Z_{1,1}, Z_{2,2}) \\ &= \{NP, Noun\} \{VP, TV\} \\ &= \{NP VP, NP TV, Noun VP, Noun TV\} \\ &= \{S\} \end{aligned}$$

Steps:

- ✓ Looking for rules of productions that generate 'NP VP, NP TV, Noun VP, Noun TV'.
- ✓ There are two productions, that is 'S \rightarrow NP VP'.
- ✓ $Z_{1,2} = \{S\}$

The construction $Z_{1,2}$ is shown in figure 3.7. As Start symbol 'S' is generated so the given input sentence is grammatically correct.

3.4.3 Dynamic-Parsing of WH-Question

Illustrating the process of dynamic-parsing of WH-Question sentence first we take an example in ATMS for Bangla language. That is:

‘আপনি কোথায় যাবেন’

The CNF form of the rule for parsing the sentence is:

$S \rightarrow NP VP / Wh VP;$

$NP \rightarrow Pronoun;$

$VP \rightarrow TV / Wh TV ;$

$Wh \rightarrow \text{কোথায়} ;$

$Pronoun \rightarrow \text{আপনি};$

$WH \rightarrow \text{কোথায়};$

$TV \rightarrow \text{যাবেন};$

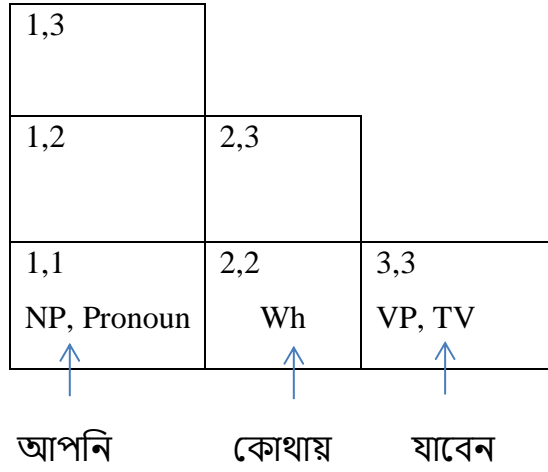


Figure 3.8: Bottom Row Construction

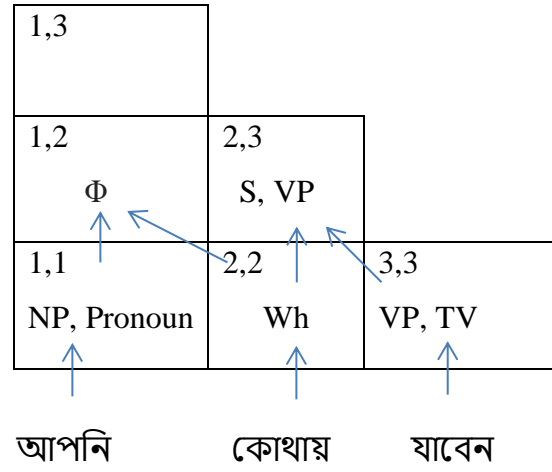


Figure 3.9: 2nd Row Construction

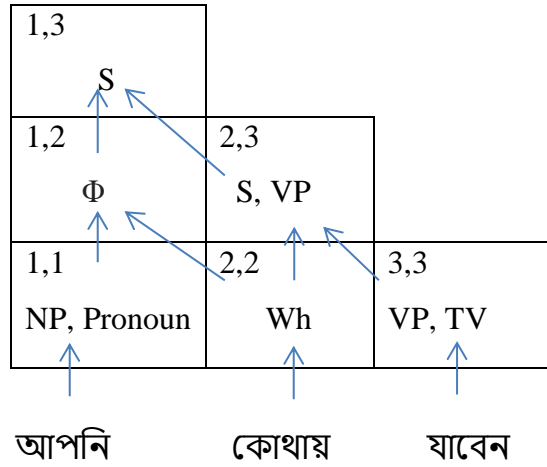


Figure 3.10: Top Row Construction

From the above equation (1) we found:

$$Z_{m,n} = (Z_{m,m} \ Z_{m+1,n}), (\ Z_{m,m+1} \ Z_{m+2,n})..... (Z_{m,n-1}, Z_{n,n})$$

$$Z_{1,2} = (Z_{m,m}, Z_{m+1,n}) = (Z_{1,1}, Z_{2,2})$$

$$= \{NP, Pronoun\} \ \{WH\}$$

$$= \{NP \ WH, Pronoun \ WH\} = \{\Phi\}$$

Steps:

- ✓ Looking for rules of productions that generate ‘NP WH, Pronoun WH’.
- ✓ There are no productions..
- ✓ $Z_{1,2} = \{ \Phi \}$

$$\begin{aligned} Z_{2,3} &= (Z_{m,m}, Z_{m+1,n}) = (Z_{2,2}, Z_{3,3}) \\ &= \{WH\} \{VP, TV\} \\ &= \{WH VP, WH TV\} \end{aligned}$$

Steps:

- ✓ Looking for rules of productions that generate ‘WH VP, WH TV’.
- ✓ There are two productions, that is ‘S, VP’.
- ✓ $Z_{2,3} = \{S, VP\}$

$$\begin{aligned} Z_{1,3} &= (Z_{m,m}, Z_{m+1,n}) (Z_{m,m+1}, Z_{m+2,n}) \\ &= (Z_{1,1}, Z_{2,3}), (Z_{1,2}, Z_{3,3}) \\ &= \{NP, Pronoun\} \{S, VP\} \cup \{ \Phi \} \{VP, TV\} \\ &= \{NP S, NP VP, Pronoun S, Pronoun VP\} \\ &= \{S\} \end{aligned}$$

As Start symbol ‘S’ is generated so the given input sentence is grammatically correct.

Steps:

- ✓ Looking for rules of productions that generate ‘NP S, NP VP, Pronoun S, Pronoun VP’
- ✓ There is one production, that is ‘S’.
- ✓ $Z_{1,3} = \{S\}$

Construction of $Z_{1,2}$, $Z_{2,3}$, $Z_{1,3}$ are shown in figure- 3.8, figure-3.9 and figure-3.10 respectively.

3.4.4 Dynamic-Parsing of Yes-No Answer Type Question

Illustrating the process of dynamic-parsing of Yes-NO-Question sentence first we take an example in ATMS in Bangla language. That is:

‘আপনি যাবেন কি’

The CNF form of the rule for parsing the sentence is:

$S \rightarrow NP VP$;

$Aux \rightarrow$ কি;

$TV \rightarrow$ যাবেন;

$Pronoun \rightarrow$ আপনি;

$NP \rightarrow Pronoun$;

$VP \rightarrow TV / TV Aux$;

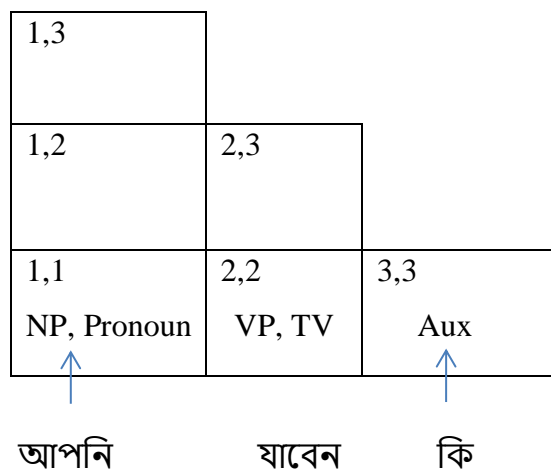


Figure 3.11: Bottom Row Construction

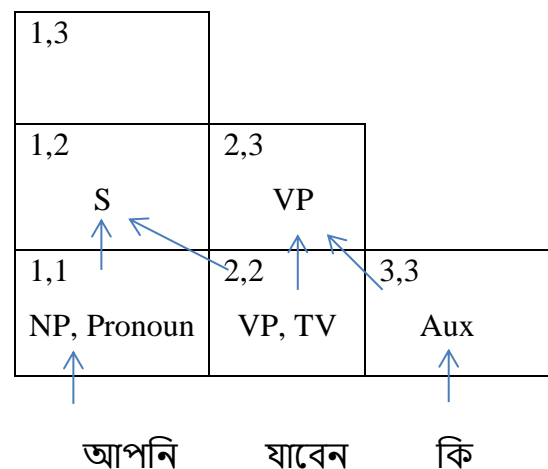


Figure 3.12: 2nd Row Construction

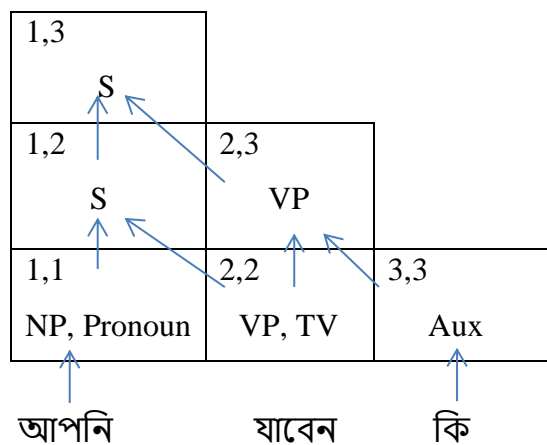


Figure 3.13: Top Row Construction

From the above equation (1) we found;

$$Z_{m,n} = (Z_{m,m} Z_{m+1,n}), (Z_{m,m+1} Z_{m+2,n}) \dots (Z_{m,n-1}, Z_{n,n})$$

$$Z_{1,2} = (Z_{1,1}, Z_{2,2})$$

$$= \{NP, Pronoun\} \{VP, TV\}$$

$$= \{NP VP, NP TV, Pronoun VP, Pronoun TV\}$$

$$= \{S\}$$

Steps:

- ✓ Looking for rules of productions that generate 'NP WH, Pronoun WH'.
- ✓ There is one production, that is 'S'.
- ✓ $Z_{1,2} = \{S\}$

$$Z_{2,3} = (Z_{1,2}, Z_{3,3})$$

$$= \{VP, TV\} \{Aux\}$$

$$= \{VP Aux, TV Aux\}$$

$$= \{VP\}$$

Steps:

- ✓ Looking for rules of productions that generate 'VP Aux, TV Aux'.
- ✓ There are one production, that is 'VP'.
- ✓ $Z_{2,3} = \{VP\}$

$$(Z_{1,2}, Z_{3,3}) = (Z_{1,1}, Z_{2,3}), (Z_{1,2}, Z_{3,3})$$

$$= \{NP, Pronoun\} \{VP\} \cup \{S\} \{Aux\}$$

$$= \{NP VP, Pronoun VP, S Aux\}$$

$$= \{S\}$$

Steps:

- ✓ Looking for rules of productions that generate ‘NP VP, Pronoun VP, S Aux.’
- ✓ There is one production, that is ‘S’.
- ✓ $Z_{1,3} = \{S\}$

As Start symbol ‘S’ is generated so the given input sentence is grammatically correct.

Construction of $Z_{1,2}$, $Z_{2,3}$, $Z_{1,3}$ are shown in figure- 3.11, figure-3.12 and figure-3.13 respectively.

3.5 Pseudocode

let us take the input sentence S which consists of z characters, that is: $a_1 \dots a_z$

let the CNF form of the grammar hold m nonterminal symbols $X_1 \dots X_m$.

let the start symbol be S_t .

let $Q[n,n,m]$ be the array consists of Booleans and initialized the elements with ‘false’.

for every $s : 1 \dots z$

for every unit production $X_v \rightarrow a_s$

set the value of $Q[I,s,v]$ to true

for every $L : 2 \dots N$ *# Determines span length*

for every $s : 1 \dots ((N-L)+1)$ *# Determines the start of the span*

for every $q : 1 \dots L-1$ *# Determines the divisions of the span*

for every production $X_b \rightarrow X_c X_d$

if $Q[q,s,c]$ and $Q[L-q,s+q,d]$ **then** fix $Q[L,s,b] = \text{true}$

if $Q[z,1,1]$ is equal to true **then**

S is recognized by the language

else

S is not recognized by the language

Chapter 4

Experiment Results and Analysis

4.1 Introduction

At the stage of the implementation of this thesis, we successfully implemented CYK parsing algorithm. We are able to handle the issue of ambiguity and left factoring if the grammar is left recursive. This parsing algorithm is faster than the other parsing techniques because the context free grammar is converted into CNF for avoiding harder calculations.

4.2 Data Collection Process

In this chapter, we have done our implementation and analysis the result. In Bangla language there are many native forms of the same sentence. But we have avoided the native form and worked on the simple form of the language that is understand and used by everyone. We have collected four type of sentences that are declarative, imperative, yes-no answer type question and WH questions and the data is collected from the various kind of people related to the domain. Here, in the output columns ‘1’ denotes that ‘Input Sentence is Correct’ and ‘0’ denotes that ‘Input Sentence is Not Correct’.

Table 4.1.1: Input Data table (Declarative Sentences)

Sentence Type: <i>Declarative</i>			
Serial Number	Input Sentence	Original output	System Output
1.	আমি গতকালের ফ্লাইট মিস করেছি	1	1
2.	শনিবার কোনো ফ্লাইট নাই	1	1
3.	আগামীকাল রাতে ঢাকার ফ্লাইট বাতিল করা হয়েছে	1	1
4.	আমি সন্ধ্যায় যেতে চাচ্ছি	1	1

5.	আমি টিকিট বুকিং করতে চাই	1	1
6.	টিকিট বুকিং করা যাবে	1	1
7.	আমি রাতের বিমানে যাব	1	1
8.	আমি ঢাকার টিকেট চাই	1	1
9.	আমি বিমানের সময়সূচী জানতে চাই	1	1
10.	আমি পরশু সকালের টিকিট চাই	1	1
11.	আমি অগ্রীম টিকিট চাই	1	1
12.	আগামীকালকের টিকিট কাটতে চাচ্ছি	1	1
13.	আমি বিকালে যাব	1	1
14.	দুই ঘন্টা পর বিমান আসবে	1	1
15.	আমার পাসপোর্ট নাই	1	1
16.	আমি ঢাকার টিকিট কাটব	1	1
17.	আমার টিকিট লাগবে কালকের	1	1
18.	আপনি যেতে পারেন	1	1
19.	আপনাকে ধন্যবাদ	1	1
20.	বিমান আসতে দুই ঘন্টা দেরি হবে	1	0
21.	আমি টিকেটটি বাতিল করতে চাই	1	1
22.	আপনি বেশি ওজন বহন করতে পারবেন না	1	1
23.	অল্প সময় পর বিমান চলে আসবে	1	1
24.	আমি বিমানে নাস্তা চাই	1	1
25.	আপনি সিটবেল্ট পরিধান করুন	1	1
26.	আপনি মাস্ক ব্যবহার করুন	1	1
27.	আপনার মালামালের বেশি ওজন হয়েছে	1	1
28.	আপনার মালামাল বেশি হয়েছে	1	1
29.	আপনি ভুল তথ্য দিয়েছেন	1	1
30.	আপনি ভুল তথ্য প্রদান করেছেন	1	1
31.	আপনার যাত্রা শুভ হোক	1	1
32.	আপনার পাসপোর্ট টি জাল পাওয়া গেছে	1	1

33.	আপনার পাসপোর্ট টি নকল পাওয়া গেছে	1	1
34.	আপনি মালামাল গ্রহণ করুন	1	1
35.	আপনার মালামাল সাবধানে রাখুন	1	1
36.	আপনার মালামাল উপরে রাখুন	1	1
37.	আপনি ভুল আসন গ্রহণ করেছেন	1	1
38.	আপনি সর্বোচ্চ ২০কেজি মালামাল নিতে পারেন	1	1
39.	আমি ব্যাগ হারিয়ে ফেলেছি	1	1
40.	আপনার সাথে টিকিট নিন	1	0
41.	আমার জরুরী টিকিট চাই	1	1
42.	আপনার মালামাল গুছিয়ে নিন	1	1
43.	আপনি আসন গ্রহণ করুন	1	1
44.	আমি টিকিট বুকিং করব	1	1
45.	আমি টিকিট বুকিং করতে চাই	1	1
46.	ফ্লাইট ছাড়তে দেরি হবে	1	1
47.	ফ্লাইট দুই ঘন্টা পরে আসবে	1	1
48.	আপনার টিকিটের মেয়াদ অতিক্রান্ত হয়েছে	1	1
49.	আপনার ভ্রমণ শুভ হোক	1	1
50.	আমাকে বিজনেসক্লাস টিকিট দিন	1	1
51.	আমাকে বিমানের সময়সূচি জানান	1	1
52.	আপনি দেরি করে ফেলেছেন	1	1
53.	আমি যাব বিকালে	0	1
54.	আপনি সাথে খাবার নিতে পারবেন না	1	0
55.	আমাকে সকালের টিকিট দিন	1	1
56.	আমাকে টিকিট দিন দুটি	0	0
57.	আপনার সাথে পাসপোর্ট রাখুন	1	0
58.	আপনার কিছু হারিয়ে গেলে আমাদের জানান	1	0
59.	আমি সাথে খাবার নিতে চাই	1	0
60.	আমি যাব না বিকালে	0	0

61.	আপনার টিকিটের মেয়াদ শেষ হয়েছে	1	1
62.	বিমান ছাড়তে দেরি হতে পারে	1	1
63.	আমার টিকিট লাগবে না কালকের	0	0
64.	বিমানের অবতরণ করতে দেরি হবে	1	0
65.	আপনি সাথে বেশি ওজন নিয়েছেন	1	0
66.	আমার সাথে ব্যাগ আছে	1	0
67.	বেশি ওজন আপনি সাথে নিয়েছেন	1	0
68.	আপনার পাসপোর্ট সাথে রাখুন	1	1
69.	আপনি ভুল তথ্য প্রদান করেছেন	1	1
70.	বিমান আসতে দেরি হতে পারে	1	1
71.	আমি টিকিট অগ্রীম চাই	0	1
72.	কোনো ফ্লাইট আজকে নেই	0	0
73.	আপনি সাথে বেশি ওজন নিয়েছেন	1	0
74.	আমি যেতে চাই না সকালে	0	0
75.	কোনো ফ্লাইট আজকে নেই	1	0
76.	আপনার শুভ বিমান ভ্রমণ হোক	0	1

Table 4.1.2: Input Data table (Yes-No-Question Sentences)

Sentence Type: Yes-No-Questions			
Serial Number	Input Sentence	Original output	System Output
1.	আমি কি ঢাকা যাওয়ার টিকেট আজকে পেতে পারি ?	1	1
2.	আমি কি এখন কোনো ফ্লাইট পেতে পারি ?	1	1
3.	আমি কি এখন কোনো ঢাকা যাওয়ার ফ্লাইট পেতে পারি ?	1	1
4.	আমি কি খুলনা যাওয়ার ফ্লাইট বাতিল করতে পারি ?	1	1
5.	আগামীকাল ঢাকা থেকে রাজশাহী কোনো ফ্লাইট হবে ?	1	1
6.	বিমানে কি কোনো ছবি তোলা যাবে ?	1	1

7.	কালকে সকালে কি ফ্লাইট আছে ?	1	1
8.	কি খাবার নিতে পারি ?	0	1
9.	আমার ফ্লাইট বাতিল হয়েছে কি ?	1	1
10.	কালকে সকালের টিকিট কি পাওয়া যাবে ?	1	1
11.	রাতের টিকিট কি পাওয়া যাবে এখন ?	1	1
12.	টিকিট কি বুকিং করা যাবে ?	1	1
13.	কাল যশোর থেকে যাওয়ার কোন ফ্লাইট আছে ?	1	1
14.	আপনি কি আজকের ফ্লাইটে যাবেন ?	1	1
15.	আগামীকালের টিকিট কি পাওয়া যাবে ?	1	1
16.	আজকের টিকিট বাতিল করা যাবে ?	1	1
17.	আজকে কি কোন শিডিউল দেরি আছে ?	1	0
18.	আপনি কি জানালার পাশের আসন চান ?	1	1
19.	আপনি কি সাথে পাসপোর্ট নিয়েছেন ?	1	0
20.	আমি কি আগামীকাল রাতের টিকেটটি ক্যান্সেল করতে পারি ?	1	0
21.	বিমানে কি নাস্তা দিবে ?	1	1
22.	বিমান কি দেরিতে ছাড়বে ?	1	1
23.	ভোরের টিকিট কি এখন পাওয়া যাবে ?	1	0
24.	আমি কি জানালার ধারের সিট টি পেতে পারি ?	1	1
25.	আপনাদের টিকিট সেবা কি অনলাইনে পাওয়া যাবে ?	1	1
26.	আপনাদের টিকিট কি ফেরতযোগ্য ?	1	1
27.	সব তথ্য সঠিক আছে কি ?	1	1
28.	আপনি কি আমাকে সিটবেল্ট বাধতে সাহায্য করতে পারবেন ?	1	1
29.	আপনি কি জানালার পাশের আসন চান ?	1	1
30.	বিমানে কি নাস্তার ব্যবস্থা আছে ?	1	1
31.	পরিবহনে বেশি সময় লাগতে পারে কি ?	1	1
32.	কোনো সময়সূচী দেরি আছে কি ?	1	1
33.	আপনার সাথে কি টিকিট আছে ?	1	0

34.	ফিরতি টিকিট কি এখন বুকিং করে রাখবেন ?	1	1
35.	আপনি কি বিজনেসক্লাস টিকিট চান ?	1	1
36.	কোনো ফ্লাইট কি দেরিতে আছে ?	1	1
37.	আপনি কি টিকিট বুকিং করবেন ?	1	1
38.	আমার ফ্লাইট কি বাতিল হয়েছে ?	1	1
39.	ঢাকা যাওয়ার টিকিট আমি কি আজকে পেতে পারি ?	1	1
40.	কালকের সকালে কি টিকিট পাওয়া যাবে ?	1	1
41.	কি যেতে পারি ?	0	1
42.	আজকের টিকিট কাটতে কি পারি ?	1	0
43.	আমি কি সাথে খাবার নিতে পারি ?	1	0
44.	কালকে কি সকালে টিকিট আছে ?	0	0

Table 4.1.3: Input Data table (WH-Question Sentences)

Sentence Type: WH-Questions			
Serial Number	Input Sentence	Original output	System Output
1.	আপনার পাসপোর্ট নম্বর কত ?	1	1
2.	আপনি কখন যেতে চান ?	1	1
3.	ফ্লাইট কখন আসবে ?	1	1
4.	আপনি কখন ও কোথায় যেতে চান ?	1	0
5.	আপনাদের মালামাল মূল্য কত ?	1	1
6.	আপনি কোন সময় ভ্রমণ করবেন ?	1	1
7.	আগামীকাল সকালে ঢাকা যাওয়ার দুইটা টিকেটের দাম কত ?	1	1
8.	ফ্লাইট কতক্ষণ পর ছাড়বে ?	1	0
9.	অবতরণের সময় কখন ?	1	1
10.	সর্বোচ্চ কত ওজন সাথে নেওয়া যাবে ?	1	0
11.	সর্বোচ্চ কত ওজন রাখা যাবে ?	1	1

12.	ব্যাগের ওজন কত ?	1	1
13.	কখন ছাড়বে বিমান ?	0	1
14.	বিমান কখন আসবে ?	1	1
15.	আমি ব্যাগ কোথায় রাখব ?	1	1
16.	আপনাদের আজকে কয়টা ফ্লাইট আছে ?	1	1
17.	আপনি কোন শ্রেণী টিকিট চান ?	1	1
18.	আপনাদের বিমান কখন আছে ?	1	1
19.	আগামীকাল রাতের খুলনা যাওয়ার টিকেট কখন পাওয়া যাবে ?	1	1
20.	ঢাকা থেকে চট্টগ্রামের শিডিউল কি ?	1	1
21.	ঢাকা থেকে সৈয়দপুরের ভাড়া কত ?	1	1
22.	বিমানের শিডিউল কি ?	1	1
23.	কখন বিমান ছাড়বে ?	1	1
24.	আপনি কোথায় যেতে চান ?	1	1
25.	আপনি কখন খুলনা যেতে চান ?	1	1
26.	আজ রাতে যশোরের ফ্লাইট কয়টায় ?	1	1
27.	শনিবার কখন ঢাকা যাবার টিকিট পাওয়া যাবে ?	1	1
28.	টিকিটের দাম কত ?	1	1
29.	কোন বিমানের টিকেট চান	1	1
30.	আপনি কোথায় যাবেন ?	1	1
31.	ওয়াশরুম কোথায় ?	1	1
32.	কত সময় ছাড়তে লাগবে ?	0	1
33.	সিট কোনটা খালি আছে ?	1	1
34.	কে যাবে এবং কোথায় যাবে ?	1	0
35.	টিকিট কখন দিতে পারবেন আপনি ?	1	1
36.	আপনার কয়টা টিকিট লাগবে ?	1	1
37.	কখন পাওয়া যাবে টিকিট ?	0	1
38.	আমি আমার ব্যাগ কোথায় রাখব ?	1	1
39.	পরবর্তী ফ্লাইট কখন ছাড়বে ?	1	1

40.	কখন যাবেন আপনি ?	0	1
41.	কোথায় যাবেন আপনি ?	1	1
42.	আজ রাতে কয়টায় ঢাকার ফ্লাইট ?	0	0
43.	আমি কখন ও কোথায় ঢাকা যাওয়ার টিকিট পাব ?	1	0
44.	আপনি টিকিট কয়টা চান ?	0	1
45.	নম্বর পাসপোর্ট কত ?	0	1
46.	সিট খালি আছে কোনটা ?	1	0
47.	ফ্লাইট আপনার কখন ?	0	1
48.	আপনাদের বিমান আছে কখন ?	0	1
49.	আপনি যেতে চান কখন ?	0	0
50.	কখন সময়সূচী আছে বিমানের ?	0	1

Table 4.1.4: Input Data table (Imperative Sentences)

Sentence Type: Imperative			
Serial Number	Input Sentence	Original output	System Output
1.	বোর্ডিং পাসটি দেখান	1	1
2.	প্লেনের সময়সূচী জানান	1	1
3.	পাসপোর্ট দেখান	1	1
4.	জানালায় পাশের সিট চাই	1	1
5.	মালামাল জমা করে দিন	1	1
6.	দুপুরে নিরামিষ খাবার দিবেন	1	1
7.	টিকিট দেখান	1	1
8.	দয়া করে যেতে দিন	1	0
9.	আজকে যাব না	1	1
10.	রাতের বিমানে যাব	1	1
11.	বিমান দেরিতে ছাড়লে আমাকে জানান	1	0
12.	বিমানের সময়সূচী জানতে চাই	1	1

13.	অগ্রীম টিকিট চাই	1	1
14.	কিছু হারিয়ে গেলে আমাদের জানান	1	0
15.	খাবার পেতে চাই	1	1
16.	একটা ঢাকার টিকিট দিবেন	1	1
17.	সিটবেল্ট বাধতে সাহায্য করুন	1	1
18.	বিমানের সময়সূচী জানান	1	1
19.	পাসপোর্ট জমা দিন	1	1
20.	সিট বুকিং করুন	1	1
21.	সিট গ্রহণ করুন	1	1
22.	প্লেন দেহেতে ছাড়বে কিনা জানতে চাই	1	0
23.	দয়াকরে সিটবেল্ট বেধে নিন	1	0
24.	সিট জানালার পাশের চাই	1	1
25.	খাবার নিরামিষ দিবেন	0	1
26.	যাব না আজকে	0	0
27.	বিমানে রাতের যাব	0	1
28.	যেতে চাই ঢাকা	0	0
29.	যেতে চাই না আজকে ঢাকা	0	0

4.3 Result Analysis

We have made an interface to check the validity of our work. The user have to put a Bangla sentence as input and using the parsing technique internally it will generates an output whether the input is in the grammar or not.

The four types of sentences are:

- Declarative Sentence
- Imperative Sentence
- Yes-no answer type sentences
- WH question

a) Declarative Sentence:

- i. আপনার মালামালের ওজন বেশি হয়ে গিয়েছে।
- ii. আপনার বিমান ভ্রমণ শুভ হোক।

Production Rules:

NP -> Pronoun Noun/ Det Noun/ Pronoun NP/ Adjective NP/ NP NP/ Noun Noun/ Noun/ NP Noun/ Pronoun NP/ NP Adjective/ Pronoun/ Pronoun Adjective Noun/ Modifier Noun/ Pronoun Modifier Noun/ Noun Adjective Noun/ Noun Modifier Noun/ Pronoun Modifier Noun/ Noun NP/ Adjective Det Noun/ Pronoun Noun Adjective Noun/ Pronoun Pronoun/ Pronoun Pronoun Noun/ Pronoun Noun Noun/ Noun Conj Noun/ Pronoun Conj Noun/ Pronoun Conj Pronoun/ Pronoun Noun Conj Noun/ Noun Conj Noun/ Pronoun Conj Noun/ Pronoun Conj Pronoun/ Pronoun Noun Conj Noun/ Adjective Noun Conj/ Adjective Noun Adjective Noun/ Noun Adjective Noun Conj/ Noun Pronoun Conj Pronoun/ Noun Pronoun Adjective/ Noun Noun Conj Pronoun/ Noun Noun Conj Pronoun/ Pronoun Adjective/ Pronoun Pronoun conj Pronoun/ Modifier Conj Adjective/ Noun Pre Noun/ Noun Pre/ AP Noun

VP -> IV TV/ Noun IV/ TV/ Adverb TV/ Adjective Noun TV/ Noun TV/ Noun Noun TV/ Adverb IV/ IV IV TV/ Adjective Noun TV/ Noun TV/ Noun Noun TV/ Adverb IV/ IV IV TV/ Pronoun IV/ Noun TV/ Adjective Noun TV/ Modifier Noun TV/ Adjective Noun IV/ Noun VP/ Noun IV TV/ Noun Adjective Noun IV/ Noun Adjective Noun IV/ Pronoun IV TV Neg/ Noun TV TV/ Pronoun TV TV/ Pronoun TV TV Neg/ TV Adjective Noun/ IV TV Adjective Noun/ TV Adjective Pronoun/ IV TV Adjective Pronoun/ Adjective Noun TV Neg/ Adjective TV

S -> NP VP

b) Imperative Sentence:

This type of sentence is used to convey request or order and most of the time subject is hidden.
For example:

- i. আসনে বসুন।
- ii. টিকিট দেখান।

Production Rules:

NP -> Noun/ Adjective Noun/ AP Noun/ Adverb Noun

VP ->TV/ IV TV/ Adverb TV

S -> NP VP/ Adverb VP

c) Yes-No Answer Type Questions:

One can answer in yes or no in this type of question. For example:

- i. আপনি কি সংরক্ষণ আসনটি করতে চান?
- ii. আপনি কি নাস্তা নেবেন?

Production Rules:

NP -> Noun Aux/ NP Aux/ Noun Aux Noun/ NP Aux Noun/ Pronoun Aux/ Pronoun Aux Pronoun/ Pronoun Aux Adjective Noun/ Pronoun Aux NP/ Noun Aux NP/ Noun Aux Adjective Noun/ Pronoun Aux Pronoun/ Pronoun Aux Pronoun Noun/ Pronoun Aux NP/ Pronoun Aux Noun/ Pronoun Aux Pronoun Noun/ Aux Noun Pre/ Aux Noun Pre Noun

VP -> TV Aux/ Noun Aux Adverb IV/ Adjective Noun Aux Adverb IV/ Noun Aux Adverb IV IV/ Noun Aux Adverb IV/ Adjective Noun Aux Adverb IV/ Noun Aux Adverb IV IV/ Noun Aux Adverb IV/ Pronoun Aux VP/ IV TV/ Noun IV/ TV/ Adverb TV/ Noun TV Neg/ TV Neg/ Noun TV Adverb/ IV TV Adverb

S->NP VP/ NP VP Aux

d) WH Questions:

- i. বিমানে উঠার দরজাটি কোথায়?
- ii. আপনি কখন যেতে চান?

Production Rules:

NP -> Wh/ Wh Noun/ Pronoun Wh Noun/ Wh Noun Noun/ Pronoun Wh NP/ Wh Noun Pronoun/ Pronoun Noun Wh/ NP Wh/ Pronoun Noun Wh/ Noun Pre Wh/ Noun Pre Noun Wh/ Pronoun Modifier Noun/ Noun NP/ Noun Pre Noun/ Noun Pre/ AP Noun

VP-> Wh Noun Pronoun IV/ Wh TV/ Adverb TV/ Noun Conj Noun IV/ Wh IV TV/ IV TV/ Noun IV/ TV/ Adverb TV

S-> NP VP/ NP Wh/ NP Wh VP/ Wh NP VP

The figure-4.1, figure-4.2, figure-4.3, figure-4.4, are outputs of our system of declarative sentence, imperative sentence, yes-no answer type question and WH question. But the figure-4.5 is the output of improper input. As the input sentence cannot reach to the start symbol according the developed context free grammar, so the system detects that the input is improper or we can say that the input sentence is not correct according to our grammar. The following figures will show us visually the parsing results. These figures are generated from the graphical user interface we used through our programming.

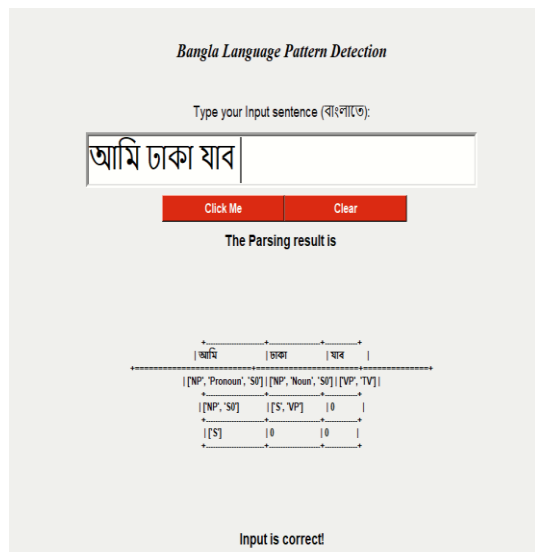


Figure 4.1: Output for Declarative Sentence

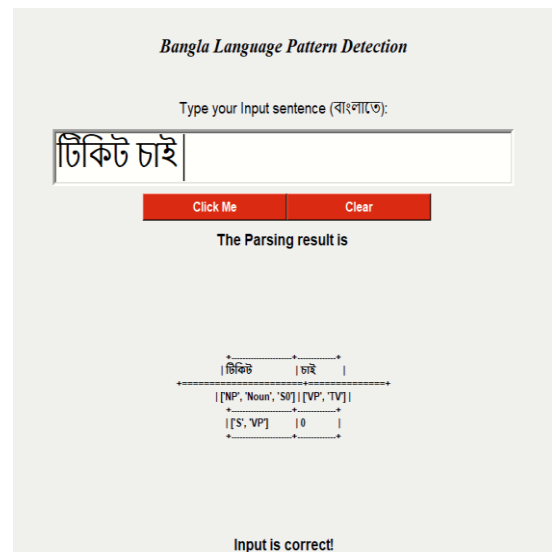


Figure 4.2: Output for Imperative Sentence

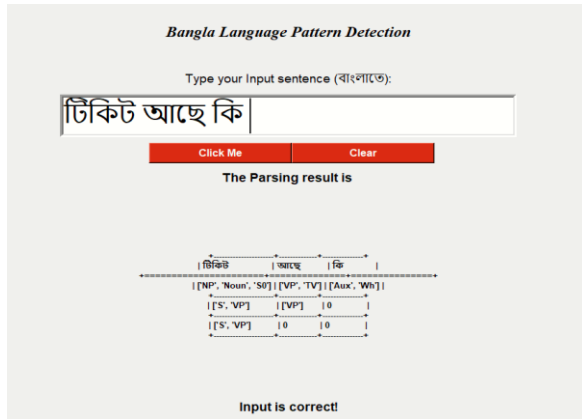


Figure 4.3: Output for Yes-No Answer Type Question

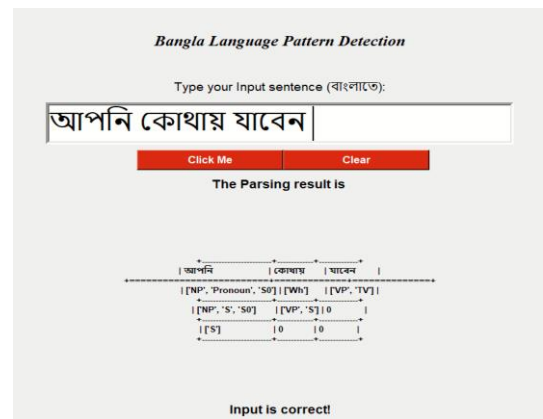


Figure 4.4: Output for WH Question

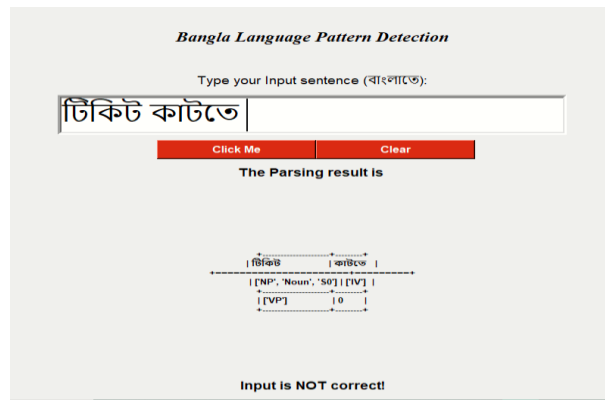


Figure 4.5: Output for improper input

Table 4.2. Performance Measurement

Serial No.	Sentence Type	Performance
1.	Declarative	80.5%
2.	Imperative	75.86%
3.	Yes-No Answer Type Questions	79.5%
4.	WH-Questions	70%

4.4 Comparison between Dynamic Parsing and Other Parsing Approach

We have studied about some parsing techniques such as predictive parsing, shift reduced parsing and related work using those parsing. There are some issues arising in predictive and shift reduce parsing.

Issues in Predictive Parsing:

- Ambiguity: Having more than one parse tree for given input string.
- Left recursive: A production of grammar is said to have left recursion if the leftmost variable of its RHS is same as variable of its LHS. A grammar containing a production having left recursion is called as Left Recursive Grammar.
- Left factoring: It consists in prefixes which are common to two or more productions.

For predictive parser we have to handle those issues otherwise parsing is not possible.

Issues in Shift Reduced Parsing:

- The Shift-Reduce Conflict is the most common type of conflict found in grammars. It is caused when the grammar allows a rule to be reduced for particular token, but, at the same time, allowing another rule to be shifted for that same token.

Reason for Applying Dynamic Programming Approach:

- At once builds the sub-solution and combined them to produce answer.
- Have no difficulty for ambiguity
- Have no issue on left recursion and left factoring
- Faster
- No issue on backtracking

Chapter 5

Conclusion

5.1 Summary

Parsing is an effective and most used technique for generating pattern. Parsing Bangla grammar is difficult as there are no fixed rule, ambiguity issue and huge collection of word and undefined structure of sentences. Having these complex issues we build our CFG base on specific ATMS domain set. We use dynamic programming where ambiguity does not create any issue. User gives a sentence as input in the system and system generates output and verifies the correctness of syntax on the basis of bottom up parsing algorithm. The CFG can cover various aspects of Bangla sentences related to ATMS domain except nontraditional format. For making it more dynamic and appropriate more production rule will be generated in future for various structures that can cover any Bangla sentence outside of the related domain.

5.2 Limitations

As Bangla language has huge number of structure and pattern so it is difficult to check all those structure and pattern. To avoid those issues we selected a domain named ATMS but still this system of checking correctness has some pitfall. It cannot check the dependencies between subject and verb. This system accepts all the subject and verb regarding to their dependencies.

5.3 Future Works

- We will design more rules for each and every possible sentence related to Air Traffic Management System.
- More effectively patterns will be modified for accurate output.

- Till now a sentence is checked whether it is grammatically right or wrong. Next a paragraph will be targeted as input to check the validity according to the grammar.
- The given input sentence accepts those sentences which is generated from the developed context free grammar but does not check any kind of dependencies between subject and verb. To get more accurate result dependencies will be checked.
- To get error free result the probabilistic parsing CYK or dependency parsing will be included.

References

- [1] K. M. A. Hasan, A. Mondal, and A. Saha, “A context free grammar and its predictive parser for Bangla grammar recognition,” In the Proceedings of ICCIT, pp. 87 – 91, 2010.
- [2] K. M. Azharul Hasan, A. Mahmud, A. Mondal, and A. Saha, “Recognizing Bangla grammar using predictive parser,” International Journal of Computer Science & Information Technology, 3(6), pp.317-326, 2011.
- [3] A. Khatun, and M.M. Hoque, "Statistical Parsing of Bangla Sentences by CYK Algorithm", "International Conference on Electrical, Computer and Communication Engineering (ECCE)", February 16-18, 2017.
- [4] M. N. Y. Ali, S. Ripon, and S. M. Allayear, “UNL based Bangla natural text conversion: Predicate preserving parser approach,” International Journal of Computer Science Issues, 9(3), pp. 259–265, 2012.
- [5] Md Asfaqul Islam, K M Azharul Hasan, Md Mizanur Rahman, “ Basic HPSG Structure for Bangla Grammar,” In the Proceedings of ICCIT, pp. 185-189, 2012.
- [6] S. Dasgupta, V. Ng, “Unsupervised morphological parsing of Bangla” Language Resources and Evaluation, 40, pp. 311–330, 2006.
- [7] Abu Zaher Md. Faridee, Francis M. Tyers, “Development of a morphological analyzer for Bangla”, In Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation, p. 43-50, 2009.
- [8] M. M. Anwar, M. Z. Anwar, and M. A. A. Bhuiyan, “Syntax analysis and machine translation of Bangla sentences,” International Journal of Computer Science and Network Security, 9(8), pp. 317–326, 2009.
- [9] Al-Mahmud, Bishnu Sarker, K M Azharul Hasan, “Parsing Bangla Grammar Using Context Free Grammar (CFG),” In Technical Challenges and Design Issues in Bangla Language Processing, IGI Global, pp. 137-154, 2013.
- [10] Rafsan Zani Rabbi, Mohammad Insanur Rahman Shuvo, K.M. Azharul Hasan “Bangla Grammar Pattern Recognition Using Shift Reduce Parser” 5th International Conference on Informatics, Electronics and Vision (ICIEV), pp.229 – 234, 2016.

- [11] Aho, A. V., Sethi, R., & Ullman, J. D. (2002). *Compilers principles, techniques and tools*. New York: Pearson Education.
- [12] Ali, M. N. Y., Ripon, S., & Allayear, S. M. (2012). UNL based Bangla natural text conversion: Predicate preserving parser approach. *International Journal of Computer Science Issues*, 9(3), 259–265.
- [13] Anwar, M. M., Anwar, M. Z., & Bhuiyan, M. A. A. (2009). Syntax analysis and machine translation of Bangla sentences. *International Journal of Computer Science and Network Security*, 9(8), 317–326.
- [14] Anwar, M. M., Shume, N. S., & Bhuiyan, M. A. A. (2010). Structural analysis of Bangla sentences of different tenses for automatic Bangla machine translator. *International Journal of Computer Science and Information Security*, 8(9).
- [15] Dasgupta, S., & Ng, V. (2006). Unsupervised morphological parsing of Bengali. *Language Resources and Evaluation*, 40, 311–330.
- [16] Faridee, A. Z. M., & Tyers, F. M. (2009). Development of a morphological analyzer for Bengali. In *Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation*, (pp. 43-50). IEEE.
- [17] Ghosh, A., Das, A., Bhaskar, P., & Bandyopadhyay, S. (2010). Bengali parsing system. Paper presented at ICON NLP Tool Contest 2010.
- [18] Haque, M. N., & Khan, M. (2005). Parsing Bangla using LFG: An introduction. *BRAC University Journal*, 2(1), 105–110.
- [19] Mehedy, L., Arifin, N., & Kaykobad, M. (2003). Bangla syntax analysis: A comprehensive approach. In *Proceedings of International Conference on Computer and Information Technology (ICCIT)*, (pp. 287-293). ICCIT.
- [20] Murshed, M. M. (1998). Parsing of Bengali natural language sentences. In *Proceedings of International Conference on Computer and Information Technology (ICCIT)*, (pp. 185-189). ICCIT.

Appendix

Serial No	Input Sentence	Sentence Type
1.	আমি গতকালের ফ্লাইট মিস করেছি	Declarative
2.	শনিবার কোনো ফ্লাইট নাই	Declarative
3.	আগামীকাল রাতে ঢাকার ফ্লাইট বাতিল করা হয়েছে	Declarative
4.	আমি সন্ধ্যায় যেতে চাচ্ছি	Declarative
5.	আমি টিকিট বুকিং করতে চাই	Declarative
6.	টিকিট বুকিং করা যাবে	Declarative
7.	আমি রাতের বিমানে যাব	Declarative
8.	আমি ঢাকার টিকেট চাই	Declarative
9.	আমি বিমানের সময়সূচী জানতে চাই	Declarative
10.	আমি পরশু সকালের টিকিট চাই	Declarative
11.	আমি অগ্রীম টিকিট চাই	Declarative
12.	আগামীকালকের টিকিট কাটতে চাচ্ছি	Declarative
13.	আমি বিকালে যাব	Declarative
14.	দুই ঘন্টা পর বিমান আসবে	Declarative
15.	আমার পাসপোর্ট নাই	Declarative
16.	আমি ঢাকার টিকিট কাটব	Declarative
17.	আমার টিকিট লাগবে কালকের	Declarative
18.	আপনি যেতে পারেন	Declarative
19.	আপনাকে ধন্যবাদ	Declarative
20.	বিমান আসতে দুই ঘন্টা দেরি হবে	Declarative
21.	আমি টিকেটটি বাতিল করতে চাই	Declarative
22.	আপনি বেশি ওজন বহন করতে পারবেন না	Declarative
23.	অল্প সময় পর বিমান চলে আসবে	Declarative
24.	আমি বিমানে নাস্তা চাই	Declarative
25.	আপনি সিটবেল্ট পরিধান করুন	Declarative

26.	আপনি মাস্ক ব্যবহার করুন	Declarative
27.	আপনার মালামালের বেশি ওজন হয়েছে	Declarative
28.	আপনার মালামাল বেশি হয়েছে	Declarative
29.	আপনি ভুল তথ্য দিয়েছেন	Declarative
30.	আপনি ভুল তথ্য প্রদান করেছেন	Declarative
31.	আপনার যাত্রা শুভ হোক	Declarative
32.	আপনার পাসপোর্ট টি জাল পাওয়া গেছে	Declarative
33.	আপনার পাসপোর্ট টি নকল পাওয়া গেছে	Declarative
34.	আপনি মালামাল গ্রহণ করুন	Declarative
35.	আপনার মালামাল সাবধানে রাখুন	Declarative
36.	আপনার মালামাল উপরে রাখুন	Declarative
37.	আপনি ভুল আসন গ্রহণ করেছেন	Declarative
38.	আপনি সর্বোচ্চ ২০কেজি মালামাল নিতে পারেন	Declarative
39.	আমি ব্যাগ হারিয়ে ফেলেছি	Declarative
40.	আপনার সাথে টিকিট নিন	Declarative
41.	আমার জরুরী টিকিট চাই	Declarative
42.	আপনার মালামাল গুছিয়ে নিন	Declarative
43.	আপনি আসন গ্রহণ করুন	Declarative
44.	আমি টিকিট বুকিং করব	Declarative
45.	আমি টিকিট বুকিং করতে চাই	Declarative
46.	ফ্লাইট ছাড়তে দেরি হবে	Declarative
47.	ফ্লাইট দুই ঘন্টা পরে আসবে	Declarative
48.	আপনার টিকিটের মেয়াদ অতিক্রান্ত হয়েছে	Declarative
49.	আপনার ভ্রমণ শুভ হোক	Declarative
50.	আমাকে বিজনেসক্লাস টিকিট দিন	Declarative
51.	আমাকে বিমানের সময়সূচি জানান	Declarative
52.	আপনি দেরি করে ফেলেছেন	Declarative

53.	আমি যাব বিকালে	Declarative
54.	আপনি সাথে খাবার নিতে পারবেন না	Declarative
55.	আমাকে সকালের টিকিট দিন	Declarative
56.	আমাকে টিকিট দিন দুটি	Declarative
57.	আপনার সাথে পাসপোর্ট রাখুন	Declarative
58.	আপনার কিছু হারিয়ে গেলে আমাদের জানান	Declarative
59.	আমি সাথে খাবার নিতে চাই	Declarative
60.	আমি যাব না বিকালে	Declarative
61.	আপনার টিকিটের মেয়াদ শেষ হয়েছে	Declarative
62.	বিমান ছাড়তে দেরি হতে পারে	Declarative
63.	আমার টিকিট লাগবে না কালকের	Declarative
64.	বিমানের অবতরণ করতে দেরি হবে	Declarative
65.	আপনি সাথে বেশি ওজন নিয়েছেন	Declarative
66.	আমার সাথে ব্যাগ আছে	Declarative
67.	বেশি ওজন আপনি সাথে নিয়েছেন	Declarative
68.	আপনার পাসপোর্ট সাথে রাখুন	Declarative
69.	আপনি ভুল তথ্য প্রদান করেছেন	Declarative
70.	বিমান আসতে দেরি হতে পারে	Declarative
71.	আমি টিকিট অগ্রীম চাই	Declarative
72.	কোনো ফ্লাইট আজকে নেই	Declarative
73.	আপনি সাথে বেশি ওজন নিয়েছেন	Declarative
74.	আমি যেতে চাই না সকালে	Declarative
75.	কোনো ফ্লাইট আজকে নেই	Declarative
76.	আপনার শুভ বিমান ভ্রমণ হোক	Declarative
77.	আপনার বিমান ভ্রমণ শুভ হোক	Declarative
78.	আমি কি ঢাকা যাওয়ার টিকেট আজকে পেতে পারি ?	Declarative

79.	আমি কি এখন কোনো ফ্লাইট পেতে পারি ?	Yes-No Question
80.	আমি কি এখন কোনো ঢাকা যাওয়ার ফ্লাইট পেতে পারি ?	Yes-No Question
81.	আমি কি খুলনা যাওয়ার ফ্লাইট বাতিল করতে পারি ?	Yes-No Question
82.	আগামীকাল ঢাকা থেকে রাজশাহী কোনো ফ্লাইট হবে ?	Yes-No Question
83.	বিমানে কি কোনো ছবি তোলা যাবে ?	Yes-No Question
84.	কালকে সকালে কি ফ্লাইট আছে ?	Yes-No Question
85.	কি খাবার নিতে পারি ?	Yes-No Question
86.	আমার ফ্লাইট বাতিল হয়েছে কি ?	Yes-No Question
87.	কালকে সকালের টিকিট কি পাওয়া যাবে ?	Yes-No Question
88.	রাতের টিকিট কি পাওয়া যাবে এখন ?	Yes-No Question
89.	টিকিট কি বুকিং করা যাবে ?	Yes-No Question
90.	কাল যশোর থেকে যাওয়ার কোন ফ্লাইট আছে ?	Yes-No Question
91.	আপনি কি আজকের ফ্লাইটে যাবেন ?	Yes-No Question
92.	আগামীকালের টিকিট কি পাওয়া যাবে ?	Yes-No Question
93.	আজকের টিকিট বাতিল করা যাবে ?	Yes-No Question
94.	আজকে কি কোন শিডিউল দেরি আছে ?	Yes-No Question
95.	আপনি কি জানালার পাশের আসন চান ?	Yes-No Question
96.	আপনি কি সাথে পাসপোর্ট নিয়েছেন ?	Yes-No Question
97.	আমি কি আগামীকাল রাতের টিকেটটি ক্যান্সেল করতে পারি ?	Yes-No Question
98.	বিমানে কি নাস্তা দিবে ?	Yes-No Question
99.	বিমান কি দেরিতে ছাড়বে ?	Yes-No Question
100.	ভোরের টিকিট কি এখন পাওয়া যাবে ?	Yes-No Question
101.	আমি কি জানালার ধারের সিট টি পেতে পারি ?	Yes-No Question
102.	আপনাদের টিকিট সেবা কি অনলাইনে পাওয়া যাবে	Yes-No Question

	?	
103.	আপনাদের টিকিট কি ফেরতযোগ্য ?	Yes-No Question
104.	সব তথ্য সঠিক আছে কি ?	Yes-No Question
105.	আপনি কি আমাকে সিটবেল্ট বাধতে সাহায্য করতে পারবেন ?	Yes-No Question
106.	আপনি কি জানালার পাশের আসন চান ?	Yes-No Question
107.	বিমানে কি নাস্তার ব্যবস্থা আছে ?	Yes-No Question
108.	পরিবহনে বেশি সময় লাগতে পারে কি ?	Yes-No Question
109.	কোনো সময়সূচী দেরি আছে কি ?	Yes-No Question
110.	আপনার সাথে কি টিকিট আছে ?	Yes-No Question
111.	ফিরতি টিকিট কি এখন বুকিং করে রাখবেন ?	Yes-No Question
112.	আপনি কি বিজনেসক্লাস টিকিট চান ?	Yes-No Question
113.	কোনো ফ্লাইট কি দেরিতে আছে ?	Yes-No Question
114.	আপনি কি টিকিট বুকিং করবেন ?	Yes-No Question
115.	আমার ফ্লাইট কি বাতিল হয়েছে ?	Yes-No Question
116.	ঢাকা যাওয়ার টিকিট আমি কি আজকে পেতে পারি ?	Yes-No Question
117.	কালকের সকালে কি টিকিট পাওয়া যাবে ?	Yes-No Question
118.	কি যেতে পারি ?	Yes-No Question
119.	আজকের টিকিট কাটতে কি পারি ?	Yes-No Question
120.	আমি কি সাথে খাবার নিতে পারি ?	Yes-No Question
121.	কালকে কি সকালে টিকিট আছে ?	Yes-No Question
122.	আপনার পাসপোর্ট নম্বর কত ?	Yes-No Question
123.	আপনি কখন যেতে চান ?	WH Questions
124.	ফ্লাইট কখন আসবে ?	WH Questions
125.	আপনি কখন ও কোথায় যেতে চান ?	WH Questions
126.	আপনাদের মালামাল মূল্য কত ?	WH Questions

127.	আপনি কোন সময় ভ্রমণ করবেন ?	WH Questions
128.	আগামীকাল সকালে ঢাকা যাওয়ার দুইটা টিকেটের দাম কত ?	WH Questions
129.	ফ্লাইট কতক্ষণ পর ছাড়বে ?	WH Questions
130.	অবতরণের সময় কখন ?	WH Questions
131.	সর্বোচ্চ কত ওজন সাথে নেওয়া যাবে ?	WH Questions
132.	সর্বোচ্চ কত ওজন রাখা যাবে ?	WH Questions
133.	ব্যাগের ওজন কত ?	WH Questions
134.	কখন ছাড়বে বিমান ?	WH Questions
135.	বিমান কখন আসবে ?	WH Questions
136.	আমি ব্যাগ কোথায় রাখব ?	WH Questions
137.	আপনাদের আজকে কয়টা ফ্লাইট আছে ?	WH Questions
138.	আপনি কোন শ্রেণী টিকিট চান ?	WH Questions
139.	আপনাদের বিমান কখন আছে ?	WH Questions
140.	আগামীকাল রাতের খুলনা যাওয়ার টিকেট কখন পাওয়া যাবে ?	WH Questions
141.	ঢাকা থেকে চট্টগ্রামের শিডিউল কি ?	WH Questions
142.	ঢাকা থেকে সৈয়দপুরের ভাড়া কত ?	WH Questions
143.	বিমানের শিডিউল কি ?	WH Questions
144.	কখন বিমান ছাড়বে ?	WH Questions
145.	আপনি কোথায় যেতে চান ?	WH Questions
146.	আপনি কখন খুলনা যেতে চান ?	WH Questions
147.	আজ রাতে যশোরের ফ্লাইট কয়টায় ?	WH Questions
148.	শনিবার কখন ঢাকা যাবার টিকিট পাওয়া যাবে ?	WH Questions
149.	টিকেটের দাম কত ?	WH Questions
150.	কোন বিমানের টিকেট চান	WH Questions
151.	আপনি কোথায় যাবেন ?	WH Questions

152.	ওয়াশরুম কোথায় ?	WH Questions
153.	কত সময় ছাড়তে লাগবে ?	WH Questions
154.	সিট কোনটা খালি আছে ?	WH Questions
155.	কে যাবে এবং কোথায় যাবে ?	WH Questions
156.	টিকিট কখন দিতে পারবেন আপনি ?	WH Questions
157.	আপনার কয়টা টিকিট লাগবে ?	WH Questions
158.	কখন পাওয়া যাবে টিকিট ?	WH Questions
159.	আমি আমার ব্যাগ কোথায় রাখব ?	WH Questions
160.	পরবর্তী ফ্লাইট কখন ছাড়বে ?	WH Questions
161.	কখন যাবেন আপনি ?	WH Questions
162.	কোথায় যাবেন আপনি ?	WH Questions
163.	আজ রাতে কয়টায় ঢাকার ফ্লাইট ?	WH Questions
164.	আমি কখন ও কোথায় ঢাকা যাওয়ার টিকিট পাব ?	WH Questions
165.	আপনি টিকিট কয়টা চান ?	WH Questions
166.	নম্বর পাসপোর্ট কত ?	WH Questions
167.	সিট খালি আছে কোনটা ?	WH Questions
168.	ফ্লাইট আপনার কখন ?	WH Questions
169.	আপনাদের বিমান আছে কখন ?	WH Questions
170.	আপনি যেতে চান কখন ?	WH Questions
171.	কখন সময়সূচী আছে বিমানের ?	WH Questions
172.	বোর্ডিং পাসটি দেখান	WH Questions
173.	প্লেনের সময়সূচী জানান	Imperative
174.	পাসপোর্ট দেখান	Imperative
175.	জানালায় পাশের সিট চাই	Imperative
176.	মালামাল জমা করে দিন	Imperative
177.	দুপুরে নিরামিষ খাবার দিবেন	Imperative
178.	টিকিট দেখান	Imperative

179.	দয়া করে যেতে দিন	Imperative
180.	আজকে যাব না	Imperative
181.	রাতের বিমানে যাব	Imperative
182.	বিমান দেরিতে ছাড়লে আমাকে জানান	Imperative
183.	বিমানের সময়সূচী জানতে চাই	Imperative
184.	অগ্রীম টিকিট চাই	Imperative
185.	কিছু হারিয়ে গেলে আমাদের জানান	Imperative
186.	খাবার পেতে চাই	Imperative
187.	একটা ঢাকার টিকিট দিবেন	Imperative
188.	সিটবেল্ট বাধতে সাহায্য করুন	Imperative
189.	বিমানের সময়সূচী জানান	Imperative
190.	পাসপোর্ট জমা দিন	Imperative
191.	সিট বুকিং করুন	Imperative
192.	সিট গ্রহণ করুন	Imperative
193.	প্লেন দেরিতে ছাড়বে কিনা জানতে চাই	Imperative
194.	দয়াকরে সিটবেল্ট বেধে নিন	Imperative
195.	সিট জানালার পাশের চাই	Imperative
196.	খাবার নিরামিষ দিবেন	Imperative
197.	যাব না আজকে	Imperative
198.	বিমানে রাতের যাব	Imperative
199.	যেতে চাই ঢাকা	Imperative
200.	যেতে চাই না আজকে ঢাকা	Imperative