Bangla Grammar Pattern Recognition Using Shift Reduce Parser

Rafsan Zani Rabbi, Mohammad Insanur Rahman Shuvo, K.M. Azharul Hasan
Dept. of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh
e-mail: rzrt.edu@gmail.com,shuvokuet09@gmail.com,azhasn@gmail.com

***Abstract*—Parser plays a very important role in computational linguistics. In this paper, here we describe a parsing technique for Bangla grammar recognition. The parser is, by nature, a shift reduce parser and constructs a parse table based on LR strategy. It takes the Context Free Grammar (CFG) of the Bangla language as input and constructs parser table from the grammar. The parse table is visited on bottom-up approach. This parser is free from the problem of the left factoring and left recursion. To avoid the inflection (BIVOKTI) of Bangla we describe a new approach. Hence only the main form of the Bangla word is stored in the repository. Our experiment shows that the scheme can detect all forms of Bangla sentences even for nontraditional forms.**

Keywords-Parser, Context free grammar, Computational linguistics, Bottom up parsing, Canonical forms, Parse table.

I. INTRODUCTION

When a language is understand by the computer, it has many practical applications, such as easy to use interfaces, machine translation and intelligent question answering systems [1].Parsing is one of the fundamental problems in computational linguistics. Parsing is a method where an algorithm is used to determine whether a given input string is grammatically correct or not for a given grammar [2].In its simplest form, the parsing problem involves the definition of an algorithm that maps any input sentence to its associated syntactic tree structure. It also involves the use of linguistic knowledge of a language to discover the way in which a sentence is structured. The question is how the linguistic knowledge is represented and can be used to understand sentences that has engaged the interest of psycholinguists, linguists, computational linguists, and computer scientists [3][4]. A parser is important for many kinds of natural language applications such grammar recognition [5][6], opinion mining [7], machine translation [8], text to speech synthesis [9], etc. We do not propose any new parsing technique for language processing we apply the idea of shift reduce parser for Bangla grammar checker.

Every valid parse tree represents a string generated by the grammar [10]. The predictive parser [11][12] is a well-known technique for its top down approach but it cannot handle the ambiguous grammar and left factoring is necessary if the grammar has left recursion. Natural language sentences are not easily parsed by programs, as there is substantial ambiguity in the structure of natural language. To avoid the problem, we describe the bottom up parser known as shift reduce parser or LR parser ("L" for left to right scan of input, "R" for rightmost derivation). Shift-reduce parsing is the simplest kind of bottom-up parsing. The LR parser starts working taking the string of the terminals and generates corresponding parse tree from the leaves to root by applying the productions in reverse. The substrings are searched from the working string which is matched in the right side of some production. The substring is then reduced. This process is repeated until the start symbol is reached. Then successful parsing is reported. The LR parser pushes "states" onto the stack in addition to the shifting tokens onto a stack. These states describe what is on the stack so far. Parsing Bangla language is a challenging task. Bangla is an order free language. Any order of the words makes syntactically correct sentence and exposes only one sense [3][4][13]. Some other challenges include two different forms namely forms of elegant, or chaste (সাধুভাষা), and colloquial (চলিত ভাষা), the symbolic vowel letters (KAR), Miscellaneous Signs (হসন্ত, ব-ফলা, ম-ফলা), inflection of words (BIVOKTI). The proposed system can handles the challenges successfully. The system can not only be used for Bangla grammar recognition but also for other natural language applications such as Semantic synthesis [14] and Opinion mining [7].

II. RELATED WORKS

There have been several parsing technique proposed for Bangla Language. But still there is no grammar checker for Bangla language. A parser has been developed in [5][13] to parse a Bangla sentence by using predictive parser. But it cannot handle the ambiguous grammar and left factoring is necessary for recursive grammars [15] proposed a top down parsing methodology for Bangla natural language sentences and shows the applicability of the phrase structure rules to parse simple sentences of Bangla. Predicate Preserving Parser

978-1-5090-1269-5/16/$31.00 ©2016 IEEE

(PPP) is described in [16] that maps Bangla text in Universal Natural Language and translates to any other natural languages. The basic structure of Head Driven Phase Structure Grammar (HPSG) has been proposed in [17] based on LKB and shows that it can handle the semantics Bangla sentence. An unsupervised morphological technique is introduced in [18] to parse Bangla sentence [19] proposed a Bangla Dependency Parser based on statistical data driven parsing system followed by a rule based post processing [2]. A rule-based Bangla parser to handle semantics as well as POS identification from Bangla sentences. A finite state technology based Bangla morphological analyzer is described in [20].A context sensitive grammar based method to analyze syntactical Bangla sentence is proposed in [8] and interpret the input Bangla sentence to English using the NLP conversion unit. It analyzes an input sentence and converts into a structural representation learning for Bangla language [15] developed for parsing Bangla sentences using LFG which is a monotonic theory of syntax. It incorporates different parallel levels of information, that can be potentially access each other. A parsing technique has been introduced to parse Bangla sentences using LFG, by designing some instruction for using the formulation of LFG rules [1] for parsing the sentences. It can work with only simple sentence and detect the syntactical problem in non-grammatical sentences [1].Hybrid Dependency Parser is also used for Bangla parsing. In this paper authors parse the language using two-stage dependency parser included data driven parser and constraint based parser and show the comparison between one and two stage parser [21]. A grammar-driven dependency parsing has been introduced for Bangla where Paninian grammatical model has been used. This method divide the complex and compound sentences in simple and then parse it and finally the parsed sentences are merged with appropriate links and Karaka labels [22]. A parser has been proposed for Bangla language which can parse all types of Bangla sentences including complex, compound, exclamatory and optative sentences along with all five categories of sentences according to Bangla intonation and handle the problem with inflection [23]. All the techniques mentioned here applied different approaches such as predictive parser [5][13], top down parser [15], HPSG structure [7], UNL based parser [16] morphological parser [17] etc. But none of them used the bottom up parser. We describe a bottom up parser in this paper.

III. INFLECTION WORD DETECTION

Bangla language analysis process suffers from the complexity of inflection word handling. Inflection word is a group of letters contains no meaning itself. But when it is attached to another word, it can modify the meaning of that word. This is called “Bivokti” in Bangla. The “Bivoktis” are divided into 7(seven) groups from 1 to 7. “Bivokti” is always added to the end of a word. The XML repository of the Bangla word does not contain the transformed form of word after adding the “Bivokti”. When the algorithm searches to find part of speech of any word, it cannot give the exact result. Therefore this is desirable to remove the “Bivokti” before parsing a sentence. We are going to propose an algorithm that can detect the root of any word by removing the inflection word. In this method the collection of “Bivokti” in Bangla language are divided into groups according to their length. It has been observed that there are 19 word inflection word are frequently used in Bangla language. They are divided into 4 group according to their respective length as shown in Table 1.

TABLE I. GROUP OF INFLECTION WORDS FOR BANGLA LANGUAGE

Length	Bivokti	Unicode Combination
1	"র", "ই", "য়",	\u09C7
2	"তে", "কে", "টি", "টা","রা",	\u09C7 + \u09B0, \u09C8 +\u09B0
3	"য়ের", "দের",	\u09C7 + \u09B0 + \u09BE
4	"গুলো", "গুলা", "গুলি", "খানা", "খানি"	No combination

The method starts searching the inflection word of smallest length. It gradually increased the length until any kind of inflection word is matched. If any inflection word of any length is found then it will be removed from end of the targeted word. For example the inflected word “পৃথিবীতে”. Here “পৃথিবী” is the root word and “তে” is the inflection word of length 2. The algorithm starts to search the inflection word length of 1. Then start to find length 2 inflection word and when “তে” is found then it is removed to form “পৃথিবী”. After removing the inflection word, the words become in the original form which is stored in the repository and is ready for search. After retrieving the part of speech from the repository, a string is constructed on the basis of the original words that contain only terminals. This string is ready to parse. This sentence then is delivered to the syntax analyzer. Fig.1 shows the parsing process as a block diagram.

Repository: An XML repository is constructed in tagged form. The Extensible Markup Language (XML) is one of the most widely used formats to store any kind tagged data. Every word has a part to speech corresponding to the position in sentence. The Bangla word is used as tag and the corresponding part of speech is stored as value of that tag. The repository contains about 1,20,000 words having no inflection.

Example 1: Consider the Bangla sentence “কৃষিতে প্রযুক্তির ব্যাবহার এখনো চোখে পড়ার মত তেমন কোন অগ্রগতি লাভ করেনি”. The division of words are as follows: W1: কৃষিতে, W2: প্রযুক্তির, W3: ব্যাবহার, W4: এখনো, W5: চোখে, W6: পড়ার, W7: মত, W8: তেমন, W9: কোন, W10: অগ্রগতি, W11: লাভ, W12: করেনি. After removing the inflection the words become in their main forms W1: কৃষি, W2: প্রযুক্তি, W3: ব্যাবহার, W4: এখন, W5: চোখ, W6: পড়া, W7: মত, W8: তেমন, W9: কোন, W10: অগ্রগতি, W11: লাভ, W12: করে. Finally retrieving the POS from the repository the sentence becomes: “**noun noun verb adj noun verb noun adv adj verb noun verb**” which is parsed by the parser.

Example2: Consider the Bangla sentence “আমি আমার দেশকে ভালবাসি”.The division of words is as follows: W1:আমি, W2: আমার,

W3: দেশকে, W4: ভালবাসি. After removing the inflection the words become in their main forms W1:আমি, W2: আমার, W3: দেশ, W4: ভালবাসি. Finally retrieving the POS from the repository the sentence becomes: "**pron pron noun verb**" which is parsed by the parser to detect whether the sentence is syntactically correct or not.

Example3: Consider the Bangla sentence "আমার দেশকে আমি ভালবাসি".The division of words is as follows: W1: আমার ,W2: দেশকে ,W3: আমি, W4: ভালবাসি. After removing the inflection the words become in their main forms W1: আমার ,W2: দেশ ,W3: আমি, W4:ভালবাসি.Finally retrieving the POS from the repository the sentence becomes: "**pron noun pron verb**" which is parsed by the parser to detect whether the sentence is syntactically correct or not.

Example4. Consider the Bangla sentence "দেশকে আমি ভালবাসি আমার".The division of words is as follows: W1: দেশকে, W2: আমি, W3: ভালবাসি, W4: আমার. After removing the inflection the words become in their main forms W1: দেশ, W2: আমি, W3: ভালবাসি, W4: আমার. Finally retrieving the POS from the repository the sentence becomes: "**noun verb pron pron**" which is parsed by the parser to detect whether the sentence is syntactically correct or not.

IV. PARSING BANGLA LANGUAGE BY SHIFT REDUCE PARSER

The shifts reduce parser works in bottom up fashion. We designed a Context Free Grammar (CFG) for shift reduce parser to develop a Bangla grammar checker. The grammar is taken as input. Then calculate the FIRST and FOLLOW of the grammar. Then the item sets of the grammar as well as the CLOSURE set for the grammar is generated. Finally the canonical set, known as LR(0) item set, is calculated by using action and goto function. Hence the parse table is constructed.

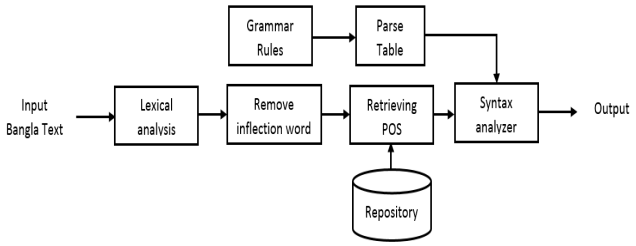

Figure 1. Model of proposed method

A. The Context Free Grammar

A Context-Free Grammar (CFG) is a set of recursive rewriting rules called productions used to generate string patterns [12]. Grammar is the first element of analysis of a language. Every language has a well-structured grammar. Like other language Bangla language stands on a well-defined grammar which is significantly complex. Mainly there are three common structure of sentence have found in Bangla language such as Simple Sentence, Complex Sentence and Compound Sentence. The simple sentence is the building block of any sentence. It usually contains one subject, one object and one verb. Example: ছাত্ররা পড়াশোনা করে।The complex sentence is consisting of one or more simple sentence and subordinate clause. Example: যেখানে বাঘের ভয় সেখানে রাত হয়।The compound sentence contains two or more simple sentence or principle clause connected by indeclinable word (অব্যয়). Example: সূর্য পূর্ব দিকে উঠে এবং পশ্চিম দিকে অস্ত যায়। Human readable form of rules of grammar cannot be recognized by computer. So here grammar rules are represented by a mathematical model.

The proposed grammar contains three non-terminals- Noun Phrase (NP), Verb Phrase (VP) and Adjective Phrase (AD). All terminals can be derived from these non-terminals. Noun phrase which is a group having a noun or a pronoun as its head. The simplest noun phrase is a single noun and it can be accompanied by modifiers, determiners (such as the, a, her), and/or complements.

TABLE II. TAG SET DESCRIPTION

SL. No	Parts-Of-Speech	symbol
1	Noun	noun
2	Pronoun	pron
3	Adjective	adj
4	Verb	verb
5	Adverb	adv
6	Conjunction	conj
7	modifier	modifier
8	Negative Particles	ptrn

Generally, a noun phrase functions as a subject, object, or complement. A verb phrase is a verb of more than one word. It includes one or more auxiliary verbs and one main verb. A verb phrase can be a syntactic unit composed of at least one verb and its dependents. An adjective phrase is a group having an adjective as its head and can be accompanied by other words such as determiners, modifiers etc. We have used Bangla word as the tag name and tagged Bangla words with their corresponding parts-of-speech (POS) tags. The (POS) tag set is stored in XML file. For example: শুভ =noun, এতে=pronoun, বাধা=verb, বকবক =adjective, বকধার্মিক =adverb, আর = conjunction, একটি =modifier, না =negative particle (ptrn). Table 1 shows the tag set description.

The CFG has been designed CFG by analyzing the natural source Bangla language such as books of famous authors, newspaper articles etc. The CFG not only considers the typical Bangla grammar rules but also the mixed and non-traditional sentences which cannot be categorized properly. A part of the CFG was used in [3][4][13] for predictive parser design Figure 2 shows the CFG.

NP -> noun ip| noun conj noun| noun pronconjpron| noun pron noun| noun pronip|noun pronadj| noun adj| noun pron pron| noun pron tp|noun noun conj pron| noun pron| pron conj pron| noun noun aw| noun noun| pron noun conj noun| pron noun adj| pron noun aw| pron noun| pron noun conjpron| pron adj| pronpron conj pron| pronpron aw| pronpron| prontp| pron| mod noun| mod adjptrn | mod adj| mod pron| mod conjadj| adjpron| adjconjadj| adj| ip| tp| xpip| xppronconjpron| xppronip| xppron noun| xppronadj| mod adj noun | xpprontp| xpadj| xp noun conjpron| xp noun aw| xp noun| xptp| xp aw| tppron| tpadj tp noun conj pron | tp pron conj pron| tp pron noun| tppronadj| tpprontp| tp noun| tp ip

VP -> noun verb| noun verb ver badj verb| noun verb verb ptrn| noun verb verb adj pron| noun verb verb adj| noun verb verb adj noun | noun verb verb noun adj verb | noun verb verb noun ptrn| noun verb verb noun adj verb| | noun verb verb noun verb aw | noun verb verb noun ptrn| noun verb verb noun pron | noun verb verb| pron adj verb| pron verb verb ptrn| noun verb verb noun aw| pron verb verb adj verb | pron verb verb adj pron| pron verb ptrn| pron verb aw| pron verb adj| pron verb pron| pron verb noun| | verb verb verb adj| pron verb verb adj noun| verb verb verb ptrn| | verb verb adj noun| verb verb adj pron| verb verb noun adj verb| verb verb noun verb ptrn| verb verb noun verb aw| verb verb ptrn| verb verb noun verb adj| verb verb noun verb pron| verb adj| verb adj verb ptrn| verb verb noun ptrn| verb adj verb noun| verb adj verb noun verb | verb adj noun| adj noun verb adj verb| adj noun verb pron| adj noun adj ptrn| adj noun adj aw| adj noun adj | adj noun verb ptrn| adjpron| conj| adjptrn

AD -> adj noun| adj pron |adj ptrn

Figure 2. CFG for Bangla grammar recognition

B. LR parsing Strategy

Input: A Bangla sentence to be parsed.

Output: accepted / rejected the sentence by the grammar.

Stack: Stack is used to store string of the form <S0G1, S1G2, GnSn>. Here S denotes a state symbol and G denotes a grammar symbol. Here, Sn is on the top of the stack.

Item Set: An LR parser takes decision of shift/reduce by maintaining states in the stack. States represent sets of "items." An LR (0) Item of a grammar G is a production of G with a dot at some position in the body. Parse table is constructed by the collection set of LR(0) items for the grammar, G denoted by I= (I0, I1, I2, ...Im).

CLOSURE: The CLOSURE (I) is the set of items constructed from I using the following two rules:

i. Every items of I is added initially to CLOSURE (I).

ii. If A-»a.Bb is in CLOSURE (I) and B->•y is a production, and then item B->.y is added to I, if it is not already there. Repeatedly apply this rule until no more new items can be added d to CLOSURE (I).

Canonical Set: The canonical set is a set of items generated by applying goto function of every terminal symbol from the initial item set. The goto function takes a state and a grammar symbol as input and produced the next state in the Deterministic Finite Automaton DFA. One collection of sets of LR(0) items, called the canonical LR(0) collection, provides the basis for constructing a (DFA) that is used to make parsing decisions.

Parse Table: It consists of two parts namely action function and goto function.

Action: The action is performed by Sn, top of the stack and x, current input symbol. The action consist of all terminal symbols. There are four operations for action i.e. Shift, Reduce, Accept and Error.

Goto: The goto is performed by using I, set of LR(0) items and G, Grammar symbol . The goto it consist of all non-terminal symbols. It helps the number terminals to be popped to replace a nonterminal and which state it goes after the transition.

Algorithm: The LR parsing starts working reading each word from input string. The procedure is summarized as follows:

1. Determine the pair < Sn, x > where Sn is the top of stack and x is the current input symbol.
2. If action [Sn,x] corresponds to shift si operation then push <x, si> to the stack .
3. If action [Sn,x] corresponds reduce operation (of the form p->q) then reduce by the right side of the production term and push the left side and its corresponding state from goto table to the stack.
4. If action [Sn,x] corresponds to accept then announce successful completion of parsing.
5. If action [Sn,x] corresponds to an empty cell of table announce error message.

S-> NP VP
NP->noun
NP->pronoun
NP->noun NP1
NP->pronoun NP2
NP->modifier noun
NP1->conjunction noun
NP1-> ε
NP2->conjunction pronoun
NP2-> ε
VP-> noun VP1
VP1-> verb VP2
VP2->verb
VP2-> ε

Figure 3. Grammar for example Example

C. An Illustrative Example

The grammar shown in Fig. 2. is sufficiently large. Hence it is not possible to show the canonical set for the whole grammar. We take a small part of the grammar shown in Fig. 3. and it's canonical set of LR(0) items are shown in Table 3.

TABLE III. CANONICAL LR(0) ITEMS FOR THE EXAMPLE GRAMMAR DESCRIPTION

I0: S'->.S S -> .NP VP NP -> .noun NP -> .pronoun NP -> .noun NP1 NP -> . pronoun NP2 NP -> .modifier noun	I4: NP -> modifier . noun I5: S -> NP VP . I6: VP -> noun . VP1 VP1 -> .verb VP2
I1: S'->S.	I7: NP -> noun NP1 . I8: NP1 ->conjunction .noun
I2: S -> NP. VP VP -> .noun VP1	I9: NP1 -> ε . I10: NP -> pronounNP2 . I11: NP2 -> conjunction . pronoun
I3: NP ->noun . NP -> noun . NP1 NP1 -> .conjunction noun NP1 -> .ε	I12: NP2 ->ε . I13: NP -> modifier noun . I14: VP -> noun VP1 . I15: VP1 -> verb . VP2 VP2 -> .verb VP2 -> .ε
I4: NP ->pronoun . NP -> pronoun . NP2 NP2 -> .conjunction pronoun NP2 -> .ε	I16: NP1 -> conjunction noun. I17: NP2 -> conjunction pronoun. I18: VP1 -> verb VP2 . I19: VP2 -> verb . I20: VP2 -> ε .

Using the grammar in Fig. 3. and the canonical LR(0) items shown in Table 3 the parse table (ACTION and GOTO) is constructed shown in Fig. 4.

	ACTION						GOTO					
	noun	**pron**	**verb**	**conj**	**mod**	**$**	**S**	**NP**	**NP1**	**VP**	**VP1**	**VP2**
0	**S3**	**S4**			**S5**		**1**	**2**				
1						**Acc**						
2	**S7**											
3	**S10**				**S9**				**8**			
4	**S13**				**S12**					**11**		
5	**S14**											
6	**R1**											
7			**S16**									**15**
8	**R4**	**R4**	**R4**									
9	**S17**											
10	**R8**	**R8**	**R8**									
11	**R5**	**R5**	**R5**									
12		**S18**										
13	**R10**	**R10**	**R10**									
14	**R6**	**R6**	**R6**									
15	**R11**											
16	**S21**		**S20**				**19**					
17	**R7**	**R7**	**R7**									
18	**R9**	**R9**	**R9**									
19	**R12**											
20	**R13**											
21	**R14**											

Figure 4. Parse Table for example grammar

V. EXPERIMENTAL RESULTS

We developed a prototype system for Bangla grammar checker using shift reduce parser. The shift reduce parser is basically used for parsing well-structured programming languages and is well studied in the literature. In this section we show the applicability of shift reduce parser for Bangla grammar checker. Fig. 5, Fig. 6, Fig. 7, Fig.8, Fig. 9 show the parsing steps for the complex, compound and simple Bangla sentences respectively.

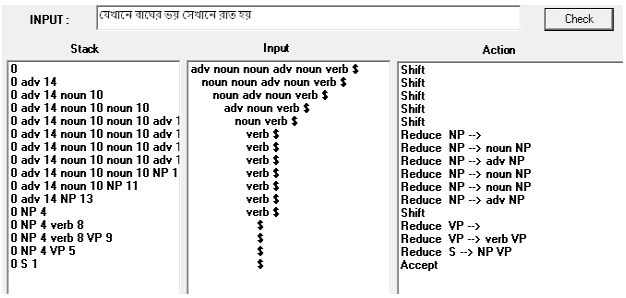

Figure 5. Parsing realization of complex Bangla sentence.

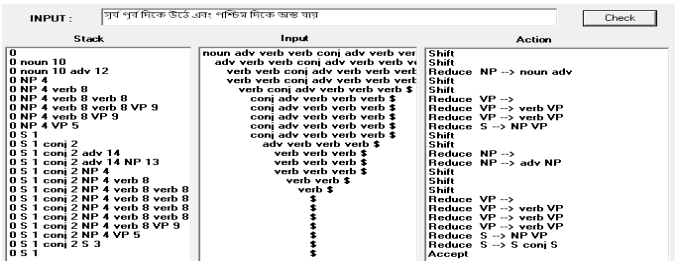

Figure 6. Parsing realization of compound Bangla sentence.

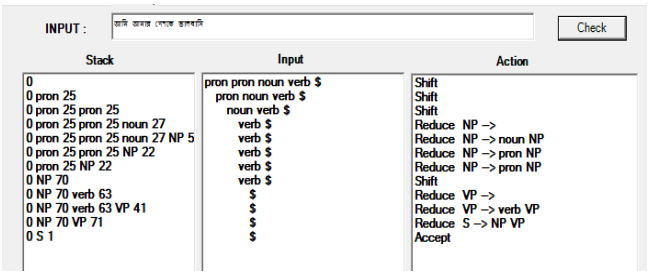

Figure 7. Parsing realization of simple Bangla sentence.

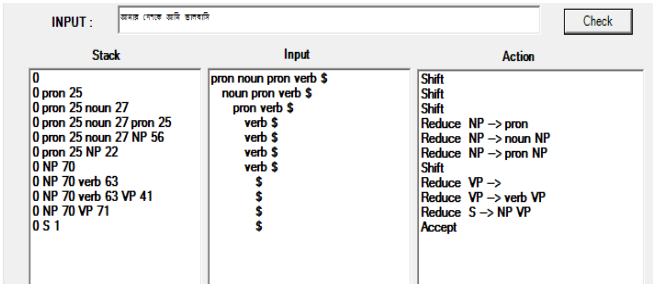

Figure 8. Parsing realization of simple Bangla sentenc

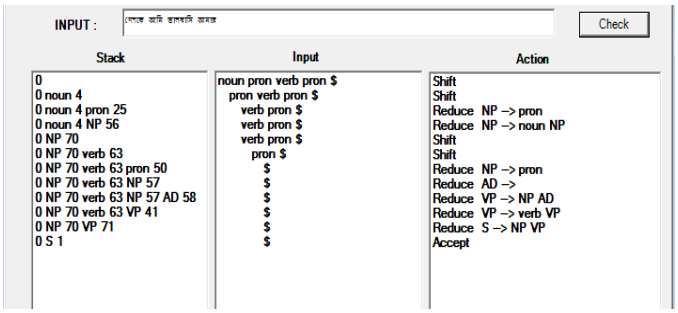

Figure 9. Parsing realization of simple Bangla sentence

Figure 10 shows parsing different kinds of Bangla sentences including simple, complex, compound and other traditional and nontraditional forms.

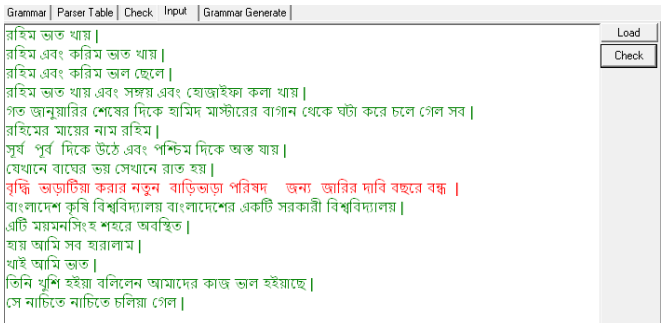

Figure 10. Parsing realization of different form of Bangla sentence

VI. CONCLUSION

Parsing natural language text is more difficult than strictly defined computer languages. The grammars for Bangla languages are often complex, ambiguous, and specified by collections of examples rather than complete formal rules. Rather it can generate by specified examples from well written texts. We defined our CFG using well written texts. Our developed prototype system is well dynamic that user can inset the grammar rules. The parser can verify a Bangla sentence is syntactically correct based on bottom up manner. The grammar can be applicable to all types of Bangla sentences even for nontraditional forms. The proposed storage of main word only saves the storage size of the repository. One important future direction is that the grammar can be applied to NPDA for Bangla grammar recognition.

VII. REFERENCES

[1] Muhammad Nasimul Haque, Mumit Khan, "Parsing Bangla Using LFG: An Introduction", BRAC University Journal, Vol. II, No. 1, , pp. 105-110, 2005.

[2] G. K. Saha, "Parsing Bengali text: An intelligent approach," ACM Ubiquity, *7*(13), pp. 1–5, 2006.

[3] 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.

[4] K M Azharul Hasan, Al-Mahmud, Amit Mondal, Amit Saha "Recognizing Bangla Grammar using Predictive Parser", In International Journal of Computer Science & Information Technology (IJCSIT), 3: 6. pp. 61-73, 2011.

[5] 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.

[6] M. M. Murshed, "Parsing of Bengali natural language sentences," In the Proceedings of ICCIT, pp. 185-189, 1998.

[7] K M Azharul Hasan, Md Sajidul Islam, G M Mashrur-E-Elahi, Mohammad Navid Izhar, "Sentiment Recognition from Bangla Text," In Technical Challenges and Design Issues in Bangla Language Processing, IGI Global, pp. 315-327, 2013.

[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] K. M. Azharul Hasan, Muhammad Hozaifa, Sanjoy Dutta, Rafsan Zani Rabbi, "A Framework for Bangla Text to Speech Synthesis," In the Proceedings of ICCIT, pp. 60-64, 2013.

[10] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," In the Proceedings of 33rd Annual Meeting of the ACL, pp. 189-196, 1995.

[11] A. V. Aho, R. Sethi, and J. D. Ullman, Compilers principles, techniques and tools, 2002.

[12] Daniel Jurafsky,James H. Martin, SPEECH and LANGUAGE PROCESSING: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Second Edition, 2009.

[13] 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.

[14] K M Azharul Hasan, Muhammad Hozaifa, Sanjoy Dutta, "Detection of Semantic Errors from Simple Bangla Sentences," In the Proceedings of ICCIT, pp 296-299, 2014.

[15] M. M. Hoque, M. M. Ali, "Context-sensitive phrase structure rule for structural representation of Bangla natural language sentences," In the Proceedings of ICCIT, pp. 615-620, 2004.

[16] 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.

[17] 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.

[18] S. Dasgupta, V. Ng, "Unsupervised morphological parsing of Bengali" Language Resources and Evaluation, *40*, pp. 311–330, 2006.

[19] A.Ghosh, A. Das, P. Bhaskar, and S. Bandyopadhyay, "Bengali parsing system, ICON NLP Tool Contest 2010.

[20] Abu Zaher Md. Faridee, Francis M. Tyers, "Development of a morphological analyzer for Bengali", In Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation, p. 43-50, 2009.

[21] Arnab Dhar, Sanjay Chatterji, Sudeshna Sarkar, Anupam Basu, "A Hybrid Dependency Parser for Bangla" In Proceedings of the 10th Workshop on Asian Language Resources, pp.55–64, COLING 2012, 2012.

[22] Utpal Garain,Sankar De, "Dependency Parsing In Bangla", In Technical Challenges and Design Issues in Bangla Language Processing, IGI Global, pp. 155-168, 2013.

[23] Lenin Mehedy, S. M. Niaz Arifin and M Kaykobad, "Bangla Syntax Analysis: A Comprehensive Approach", In the Proceedings of ICCIT, 2013.