Designing a Bangla parser using non-deterministic push down automata

Conference Paper · February 2017

DOI: 10.1109/ECACE.2017.7912970

4 authors:

Md Mostafizur Rahman
RMIT University
58 PUBLICATIONS **170** CITATIONS
SEE PROFILE

Md. Abdulla-Al-Sun
Khulna University of Engineering and Technology
1 PUBLICATION **1** CITATION
SEE PROFILE

K. M. Azharul Hasan
Khulna University of Engineering and Technology
69 PUBLICATIONS **317** CITATIONS
SEE PROFILE

Mohammad Insanur Rahman Shuvo
Khulna University of Engineering and Technology
6 PUBLICATIONS **18** CITATIONS
SEE PROFILE

Designing a Bangla Parser using Non-Deterministic Push Down Automata

Md. Mostafizur Rahman, Md. Abdulla-Al-Sun, Mohammad Insanur Rahman Shuvo, K.M. Azharul Hasan
Dept. of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh
e-mail:mostafizur.cse21@gmail.com,sunkuet02@gmail.com,shuvokuet09@gmail.com,azhasn@gmail.com

***Abstract*- The goal of language processing is to make machines be able to read human language comprehensively and use human language to communicate with human beings. To achieve this goal, the first step is to parse the sentence structure correctly. In this paper, we propose a parsing technique for Bangla Language based on Non-Deterministic Push-Down Automata (NPDA). The NPDA parser takes the Context Free Grammars (CFG) of Bangla Language for preprocessing. The NPDA is efficient especially when large number of CFGs needs to be processed. The predictive parser needs to generate parse table from the CFG if the number of CFGs are large then it is difficult to construct such a big parse table. The NPDA does not require constructing any parse table and it can process the CFG directly. The parser can detect Bangla sentences of all forms whether they are syntactically and grammatically correct or not. Finally, we compare these two parsers with their complexity and efficiency issues. Comparing with these important issues, the NPDA parser gives the better result than predictive parser. Sufficient examples and figures are described to explain the parsing idea.**

Keywords- Parser, Context Free Grammar, Predictive parser, NPDA, CNF, Left factoring, Parse table, Bangla Language processing, Unit production.

I. INTRODUCTION

Language is most important element in human communication and technology after the existence of the human being. Natural language processing (NLP) is the ability of a computer program to understand human speech as it is spoken. In computer science, language processing is one of the major fields of artificial intelligence technology. Its goal is to make machines be able to read human language comprehensively and use human language to communicate with human beings. NLPis one the major field with enormous application such as translation from one language into another, retrieval of information from databases, human/computer interaction, and automatic dictation [16]. A primary success of programming language study was the building of parser makers, tools that offer a brief, declarative method to solve the universal problem of parsing [17].Parsing is one of the basic problems in NLP [2]. Parsing is a method to analyze the structure of a sentence to detect whether it is grammatically correct or not [12][20]. Parsing is an algorithm that maps a sentence to its corresponding syntactic tree structure. A parser analyzes the sequence of symbols presented to it based on the grammar [5]. A parser gives a structural description on the input sentence by applying a given computational grammar [14]. Pushdown automata produce the main fundamental concept for the parsing of programming languages. The performance of an automata can be evaluated by how easily the can handle non determinism which is the capability of a computational machine to find a solution and then validate it [18].

Parsing natural language is difficult because of its complexness, it is highly ambiguous as well as specified by collections of examples rather than complete formal rules along with punctuation is used much more sparingly [2]. For Bangla language it is more difficult than other language because we have a larger number of words and same word can be in many forms by adding 'Bivokti', 'Prokiti-Prottoy' etc. For a syntax based grammar checking the sentence is completely parsed to check the correctness of it. If the syntactic parsing fails, the text is considered incorrect. On the other hand, for statistics based approach, Parts Of Speech (POS) tag sequences are prepared from an annotated corpus, and hence the frequency and the probability [5]. The predictive parser [4][5] 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 [1]. Generally Parsing can be divided into two major types: Top-down parsing and Bottom up parsing [3]. But in this paper we apply a different approach for Bangla language parsing by using Nondeterministic Push down automata (NPDA). In this technique we follow the steps including taking CFG for Bangla language, remove left recursion from grammar, converting left factored the null and unit production, then convert the CFG into CNF (Chomsky normal form) and then implement NPDA parser for Bangla. .

II. RELATED WORK

There are many parsing technique available for Bangla language. A parsing technique has been proposed in [1] for Bangla grammar recognition by using shift reduce parser where the parse table is accessed on bottom up approach. It can handle the left factoring and left recursion problem and also introduce a new technique for avoiding the inflection (BIVOKTI) word [1] and it can handle all 3 form of Bangla sentences. A LFG (Lexical Functional Grammar) for parsing Bangla has been proposed in [2] which offers set of

instructions for using the formulation of LFG rules for parse Bangla sentences. This technique can parse some simple Bangla sentences correctly as well as find whether it is a non-grammatical sentence. NPDA for the English Language has been implemented in [3] to modernize Context Free Grammar (CFG) for English language along with refurbish into NPDA. It is capable of parsing legitimate English language sentences by using push down stack and input tape for recognizing English language sentences [3] where Context Free Grammar is converted to Chomsky Normal Form (CNF). A parser has been introduced in [4][5] where they delineates a Context Free Grammar (CFG) for Bangla language as well as can parse Bangla sentences using predictive parser to detect whether it is syntactically correct or there exist syntactical mistakes of Bangla sentences when there is no entry for a terminal in the parse table [5]. Both of this paper [4][5] follow top down parsing method and use left factoring to avoid left recursion of the CFG. A Predicate Preserving Parser (PPP) is proposed in [6] which can transform Bangla text into Universal Natural Language (UNL) and translate to any other natural language that can help to develop a universal language translation method. A method has been proposed to analyze Bangla sentence syntactically using context-sensitive grammar rules for all form of Bangla sentences along with converts the Bangla sentence into English with the help of NLP conversion unit in [7]. A structure has been addressed in [8] to recognize Bangla grammar for detecting syntactically correct Bangla sentences and declining the incorrect sentences using Head-Driven Phrase Structure Grammar (HPSG). An approach has been introduced to parse all types of Bangla sentences as well as all five categories of sentences according to Bangla intonation using context-free grammar rules in [9] which also pay attention in decomposing inflection verb to extract information from inflection. An implementation of LFG-based parsers for Indian languages and especially for Bangla (Bengali) has been introduced in [10] which is based on a delayed evaluation of syntactic encoding schema. A grammar-driven dependency parsing has been proposed for Bangla by using Paninian grammatical model in [11]. In this paper, Complex and compound sentences portioned into simple sentences and then parse these sentences by using the Karaka demands of the Demand Groups (Verb Groups) and parsed sentences are combined with appropriate links and karaka labels finally. Bangla language is parsed using Hybrid Dependency Parser in [12] where the language is parsed using two-stage dependency parser included data driven parser and constraint based parser as well as compare among one and two stage parser. An open-source morphological analyzer for Bengali Language using finite-state technology has been addressed in [13]. A parser has been proposed in [14] to identify part of speech (POS) of Bengali lexicons in Bengali (or Bangla) sentences as well as annotating semantics information. Basic structure of Head Driven Phase Structure Grammar (HPSG) has been introduced in [8] based on LKB where it can handle the semantics Bangla sentence. An unsupervised morphological technique is proposed to parse Bangla sentence in [15]. A method has been proposed that discuss PI-(push-input-) PDA and PI-LIG without transition rule and a bottom-up method of parsing for LIGs, where the stack symbols are removed at the first stage of the parsing [19].All of the methods discussed here use Shift Reduce parser [1], predictive parser [4][5], HSPG Structure [8] , LFG [2], UNL based parser [6], morphological parser [13][15] etc. Only one method use NPDA to parse English Language [3], but none of them used NPDA for Bangla language.

III. EFFICIENT PARSING TECHNIQUES FOR BANGLA GRAMMAR RECOGNITION

A. Parsing By Non-deterministic Push-Down Automata

Non-deterministic Push-Down Automata (NPDA) is a stack implemented top down parsing technique. NPDA parsing technique can recognize all context-free languages. Non-deterministic Push-Down Automata (NPDA) can be used for Bangla language parsing. In many cases it gives better result than Predictive Parser and Shift Reduce Parser. Also non-deterministic PDA gives better result than deterministic PDA because non-deterministic PDA checks all the grammar rules whereas deterministic PDA checks few grammar rules. Table I shows a comparison of the NPDA and predictive parsing.

TABLE I. COMPARISON OF PREDICTIVE AND NPDA PARSING TECHNIQUES

	Predictive Parser	**NPDA**
Memory	Requires more memory for large grammar. Memory loss in parse table is high.	Stack may become full for large grammar.
Time	Requires more time as compare to NPDA as it calculates FIRST and FOLLOW and generate Parse Table.	Requires less time as compare to Predictive Parser.
Recognizing Capabilities	Slow as compare to NPDA	Faster than predictive parser.
Speed	For large Grammar, it will be slow.	For large grammar, it will be slow as backtracking needed.

B.Methodology

For recognizing Bangla language with NPDA parser, first we take the grammars which are not left recursive. If grammars are left recursive then we may have to compute same productions again and again. The grammars should be left factored to

restrict non-terminals to derive several production rules with same terminal symbols. Then the CFGs need to convert to Chomsky Normal Form (CNF). The cause of converting the CFGs into CNF are:

1. If there contains any null productions then it may cause to failure to parse or lose of computation.
2. If there are any unit productions then the computation is also increase or the one step more for parsing.
3. After removing null productions and unit productions the CFGs are converted in CNF for parsing Bangla Languages using NPDA.

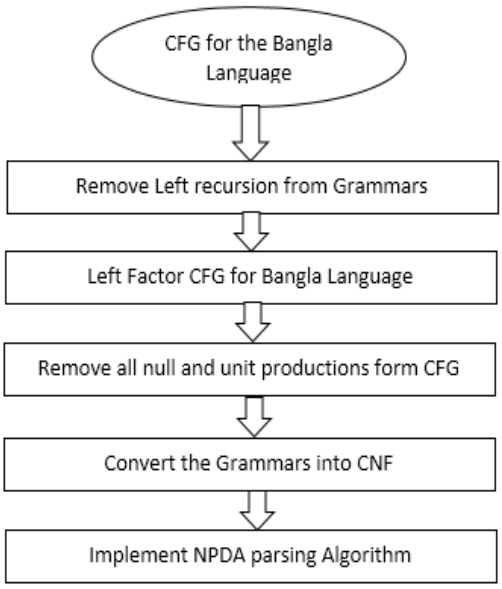

Figure 1. Steps of Bangla Language NPDA parsing

We prefer to design a parser which can recognize Bangla sentences whether it is syntactically correct or not using Non-Deterministic Push-Down Automata. Figure 1 shows the steps to convert Bangla language Context Free Grammar (CFG) into Nondeterministic Pushdown Automata (NPDA).
The first step for this conversion is to make left factored CFG for the Bangla Language which doesn't contain any null productions. Second is to kill all unit productions from the CFG and in the third step convert this CFG into CNF, this CNF is capable for the NPDA. Finally, run the algorithm with these grammars to parse the sentence whether it is grammatically correct or not. We explain the NPDA parsing by constructing some examples as follows.

Suppose three Bangla Sentences "রহিম এবং করিম ভাল ছেলে", "একটি ছেলে বই পড়ছে", "আমি তোমাকে ভালবাসি". The respective parts of speeches of the three sentences are shown below

রহিম - noun	ছেলে - noun	পড়ছে - verb
এবং - conj	একটি - adv	আমি - pron
করিম - noun	ভালবাসি - verb	তোমাকে - pron
ভাল - adj	বই - noun	

So, the overall grammars of those three sentences are "noun conj noun adj noun", "adv noun noun verb" and "pron pron verb". We design the below CFG which will accept the above three Bangla sentences.

C.Removing left recursion and Null productions

The left factored Bangla Language CFGs are given below after eliminating left recursion. Null productions are also removed from the CFGs. After removing left recursion and Null production the left factored grammar is shown in Figure 2 and Figure 3 respectively.

S→NP VP
NP→NP1|adv NP1|noun NP2|pron NP2
VP→NP1 VP1|adj NP1
NP1→noun|pron
NP2→conj NP3
NP3→noun
VP1→verb|VP2 VP3
VP2→verb
VP3→verb

Figure 2. The left factored grammars for NPDA

D.Removing Unit productions

A → B are said to be unit production if A and B are two non-terminals. In the above CFGs there is only one unit production

- NP → NP1.

This can be replaced by:

- NP → noun
- NP → pron

Figure 3 shows the grammar after removing unit production.

S→NP VP
NP→noun |pron |adv NP1|noun NP2|pron NP2
VP→NP1 VP1|adj NP1
NP1→noun|pron
NP2→conj NP3
NP3→noun
VP1→verb|VP2 VP3
VP2→verb
VP3→verb

Figure 3. Grammars for NPDA after removing unit productions

E. Converting to Chomsky Normal Form (CNF)

For converting the CFG to CNF the below two rules followed:

- One non-terminal → Exactly two non-terminals
- One non-terminal → One terminal

So, the final CNF is shown in Figure 4.

S→NP VP
NP→ noun |pron |AD NP1|NP1 NP2
VP→NP1 VP1|AJ NP1
NP1→noun|pron
NP2→CON NP3
NP3→noun
VP1→verb|VP2 VP3
VP2→verb
VP3→verb
CON→conj
AJ→adj
AD→adv

Figure 4. CNF for Bangla Grammars for NPDA

F.Bangla Language Parsing NPDA Algorithm

Nondeterministic Pushdown Automata (NPDA) uses push down stack and input string for perceiving a sentence of a language. NPDA includes the elements namely a set of input symbol, stack, a number of actions such as Start, Push, Pop, Accept, Reject etc.[3].

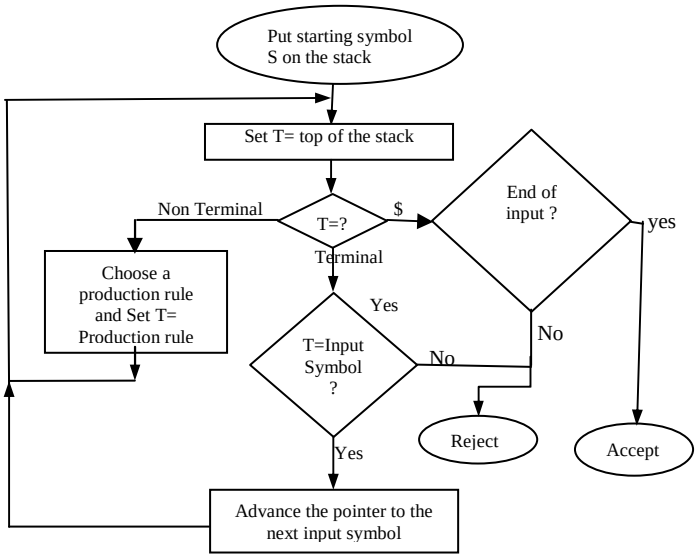

Figure 5. Flowchart of the algorithm of NPDA Parsing

The NPDA parsing needs the input sentence to be converted into its corresponding Parts Of Speech (POS).This POSes are the terminal symbols such as noun/pronoun etc (See Fig. 2). The process starts by putting the start symbol S on the stack. Then it selects the symbol which is now on the top of the stack (Let **a**). This symbol may be a Non Terminal or Terminal or $ ($ is the end marker). If **a** is a Non Terminal symbol then it choose a production rule and push the production rule to the top of the stack. And then the same process continues again. If **a** is a Terminal symbol then pointer to input string is advanced to the next input symbol. So by applying this process if we can reach to the $ symbol finally then we can accept the sentence as syntactically correct otherwise we reject the sentence. The overall process is shown in Figure 5 by using a Flow chart. Table 1 shows an example of parsing "আমি তোমাকে ভালবাসি" by NPDA stack after getting the POS of the sentence. The respective POSes are "pron pron verb"

TABLE I. EXAMPLE OF PARSING "আমি তোমাকে ভালবাসি"

States	Stack	Input
Start	$	pron pron verb $
Push S	S	pron pron verb $
Pop S	$	pron pron verb $
Push VP	VP	pron pron verb $
Push NP	VP NP	pron pron verb $
Pop NP	VP	pron pron verb $
Read NP->pron	VP	pron verb $
Pop VP	$	pron verb $
Push VP1	VP1	pron verb $
Push P1	VP1 P1	pron verb $
Pop P1	VP1	pron verb $
Read P1→pron	VP1	verb $
Pop VP1	$	verb $
Read VP1→verb	$	$
Accept	$	$

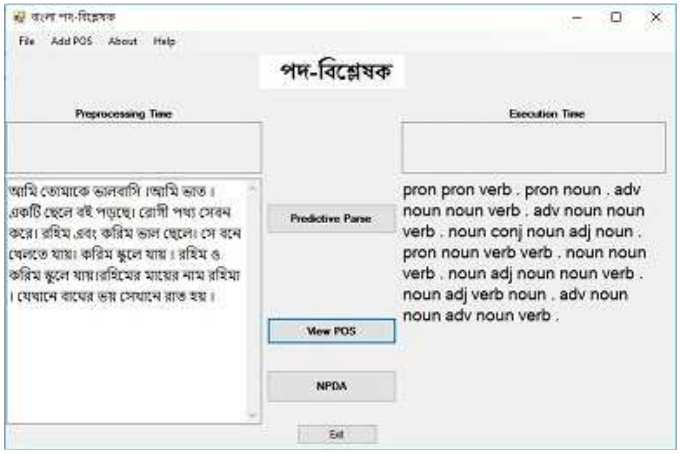

Figure 6. POS view of PODBISLESOK

IV. EXPERIMENTAL RESULTS

We have developed a prototype system for Bangla parser namely বাংলা পদ-বিশ্লেষক (BANGLA PODBISLESOK) in C++. It contains both NPDA parsing and predictive parsing scheme. Inside the PODBISLESOK there is a POS (Parts Of Speech) tagger that contains 1,20,000 Bangla words. The POS is a XML file each of the words is stored in the format

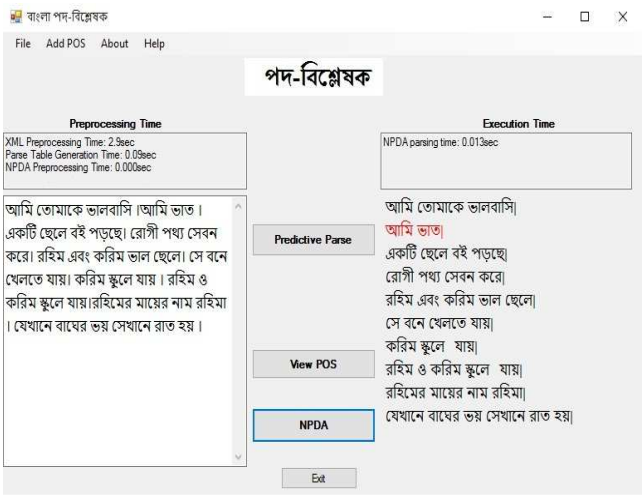

Figure 7. Checking paragraph with NPDA parsing

<word>POS</word>. Figure 6 shows a POS view of input text of PODBISLESOK. If a word is not available in the POS tagger, the GUI of the system (বাংলাপদবিশ্লেষক) has the option to add the word (See Fig. 6). It increases the reliability of the system. Figure 7 shows the parsing GUI by NPDA. It also shows the time requirement for parsing a paragraph which includes simple, complex, compound sentences as well as nontraditional sentences. The time for predictive parser has been calculated by adding preprocessing time and execution time. The system can detect which sentence is syntactically correct or not as well as indicting the grammatically wrong sentence by using red color.

TABLE II. PERFORMANCE MEASURE

Paragraph	Predictive Parser	NPDA Parser
3 lines	Preprocessing Time +Parsing Time =3.8 sec+0.001sec =3.801 sec	Preprocessing Time +Parsing Time =0 sec+0.005sec =0.005sec
10 lines	Preprocessing Time +Parsing Time =3.8 sec+0.005sec =3.805 sec	Preprocessing Time +Parsing Time =0 sec+0.013sec =0.013sec
16 lines	Preprocessing Time +Parsing Time =3.8 sec+0.008sec =3.808 sec	Preprocessing Time +Parsing Time =0sec+0.021sec =0.021sec
84 lines	Preprocessing Time +Parsing Time =3.8 sec+0.038sec =3.838 sec	Preprocessing Time +Parsing Time =0sec+0.059sec =0.059sec

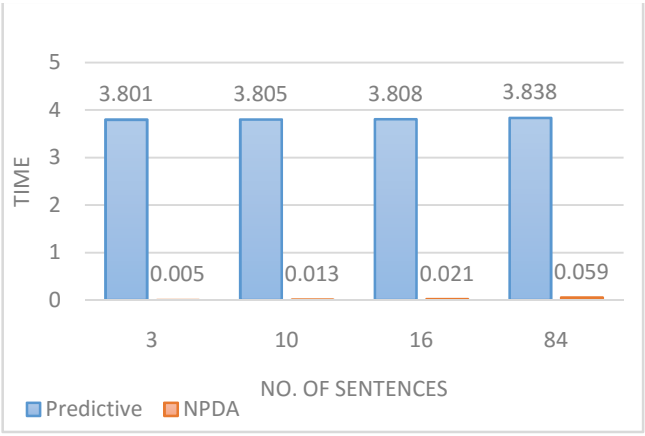

Figure 8. Running time comparison of Predictive and NPDA parsing techniques

Table II shows the performance of the parsing techniques. NPDA Parser require less time than Predictive Parser. This is because, the predictive parser has to generate a parse table. The predictive parser also requires to calculate FIRST and FOLLOW set for large number of CFGs. The Preprocessing Time in predictive parsing also includes XML preprocessing time plus parse table generation time. But NPDA does not need something like that. So overall running time for parsing is less in NPDA than Predictive parser. The running time comparison of the two parsing techniques is shown in Figure 8.

V. CONCLUSION

We develop a Bangla parser using NPDA. NPDA is efficient specially when there are a large number of grammars exist for a system. Bangla language has a long history and it has huge number of grammars. The developed Parsers can detect the grammatical and syntactical error of sentences successfully. Our POS tagger has about 1,20,000 words that correctly identifies the parts of speech of a sentence. The parser is efficient comparing to predictive parser as the NPDA does not require creating a large parse table. The parse table needs huge preprocessing works. The parsers can be applicable to any types of sentences including simple, complex and compound. One more thing is that, the parser can detect the main words having “Bivokti”. We compared our new approach with the previous one and show that the NPDA based parser works better. One important future work can be to design a more efficient algorithm for parser by combining the bottom-up and top-down approach together.

VI. REFERENCES

[1] Rafsan Zani Rabbi, Mohammad Insanur Rahman Shuvo, K.M. Azharul Hasan “Bangla Grammar Pattern Recognition Using Shift Reduce Parser”,In International Conference on Informatics, Electronics & Vision, 2016.

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

[3] Madiha Khurram Pasha and M. Sadiq Ali Khan “To Design a English Language Recognizer by using Nondeterministic Pushdown Automata

(ELR-NPDA)",International Journal of Computer Applications, Volume 105 – Number 1, 2014.

[4] K. M. Azharul 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] P. Sengupta and B. B. Chaudhuri, "A Delayed Syntactic Encoding-based LFG Parsing Strategy for an Indian Language – Bangla" Computational Linguistics, 23(2), pp. 345-351, 1997.

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

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

[13] 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.

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

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

[16] 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.

[17] Chinawat Isradisaikul and Andrew C. Myers, "Finding Counter examples from Parsing Conflicts". In Proc. of ACM Conf. on ProgrammingLanguage Design and Implementation (PLDI), pp. 555–564, 2015.

[18] Thomas A. Henzinger and Jean-François Raskin, "The Equivalence Problem for Finite Automata"In Communication Of The ACM, Vol. 58 , No. 2, pp. 86, 2015.

[19] Katsuhiko Nakamura and Keita Imada, "Eliminating Stack Symbols in Push-DownAutomata and Linear Indexed Grammars" In LATA 2013, LNCS 7810, pp. 444–455, 2013.

[20] 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,2013.