SUST Studies, Vol. 12, No. 1, 2010; P:99-105

Transformational Generative Grammar for Various Types of Bengali Sentences

Mohammad Reza Selim[1] and Muhammed Zafar Iqbal[1]

[1]*Department of Computer Science and Engineering, Shahjalal University of Science and Technology, Sylhet, Bangladesh*

Emails: *{selim, mzi}@sust.edu*

Abstract

In this paper, we analyze the syntax of various types of Bengali sentences and design transformational generative grammar rules for them. Designing machine-processable grammars to recognize and generate various types of Bengali sentences is an important step and prerequisite to develop Bengali natural language applications. However, although few works for designing grammars are found, they mainly deal with on assertive sentences. Besides, those grammars are phrase structure grammars and can only work on strictly formal sentences. Many natural language sentences cannot be parsed using only such grammars. In this paper, we design a transformational generative grammar which, in conjunction with phrase structure grammar, is used to generate or recognize other types of Bengali sentences. It is applicable for many usable sentences that cannot be parsed using only phrase structure grammars.

Keywords: Deep Structure, Surface Structure, Phrase Structure Grammar, Transformational Generative Grammar.

1. Introduction

There are several types of sentences in Bengali, e.g., Assertive, Interrogative, Imperative, Optative, Conditional, Dubitative and Interjective [1]. Each type of sentences has different *surface structure* (SS), i.e., real/usable form, although their syntactic structures, called the *deep structures* (DSs) [2, 3, 4, 6, 12, 13], are same. In order to generate or recognize a natural language sentence by a computer, its syntactic structure must be identified. In addition, grammars are necessary to map the syntactic structure to its surface form. Syntactic quantity of a sentence is represented using Phrase Structure Grammar (PSG) [2, 3, 4, 5, 6, 12, 13] while the surface form is represented using Transformational Generative Grammar (TGG) [2, 3, 4, 6, 12, 13].

PSGs use *phrase structure rules* (PSRs) [2, 3, 4, 6, 12, 13] to generate the DS of a sentence. However, a DS may not be a usable form. All sentences of a language cannot be generated using only PSGs. The usable form of a sentence, i.e., SS, can be generated using the special type of grammar TGG. The rules used in TGG are called *transformational rules* (TRs) [2, 3, 4, 6, 12, 13]. For example, the Bengali interrogative sentence (surface form) বাংলা ভাষা কি আমদের অনেক কিছু দেয়নি? can not be generated using the PSG only. However, in addition to PSG, if we use TGG, it is possible to generate such sentences. The PSG can generate the DS বাংলা ভাষা আমদের অনেক কিছু দিয়েছে only. From this DS, TRs can generate the interrogative sentence, i.e., the SS, বাংলা ভাষা কি আমদের অনেক কিছু দেয়নি?.

All most all the research works on Bengali syntax analysis consider assertive sentences only [5, 7, 8, 9, 10, 11]. No significant work has been done on other type of sentences except the preliminary analysis in our previous work [6]. The scope of this paper is to design TRs to generate or to recognize various types of Bengali sentences. Although there are five types of sentences in Bengali, we limit our discussion only on the mostly used types of sentences, i.e., assertive, interrogative and imperative sentences. In Bengali, there are simple, complex and compound sentences. Although purely simple sentences do not dominate a language, they are the building blocks of other types of sentences. Therefore, we consider simple sentences only in this paper.

The paper is organized as follows. In section 2, we design the grammar rules, specially the TRs, to generate the SSs from DSs for various types of sentences. Section 3 presents an algorithm for the reverse process, i.e., to generate the DSs from the SSs. In section 4, we present a discussion on our work. Finally, section 5 concludes our work.

2. Grammar for Different Types of Bengali Sentences

We propose here that all sentences in Bengali, whatever their types are, have same DS. The type of a sentence is identified by an abstract symbol, which has existence only in DS. The abstract symbol activates one or more TRs in order to generate the SS. Thus we can generate different types of sentences. In the next subsections, we design the TRs for different types of sentences. To explain our work, we use the PSG rules designed in [6].

2.1 Assertive-Affirmative Sentences

PSR 1: **S → NP + VP**

[**Abbreviations:** S: Sentence, NP: Noun Phrase, VP: Verb Phrase]

The syntactic structure of assertive-affirmative sentences can be represented by the above PSR 1. For this type of sentences we need no TRs because the DSs and the SSs for such sentences are same. As an example, let's generate the sentence আমার বাবা ঢাকায় থাকেন using the above PSR 1. Fig. 1 shows the syntax tree, i.e., the DS.

2.2 Assertive-Negative Sentences

In Bengali negative sentences, an indeclinable word (অব্যয়) না is used to express the negativity. The position of this word is at the end of the sentence [2]. For example, in the sentence আমি সভায় উপস্থিত ছিলাম না, the position of না is at the end. However, in the sentences like আমি বিশ্ববিদ্যালয়ে যাইনি, there is no না. But this sentence is same as আমি বিশ্ববিদ্যালয়ে গিয়েছি না.

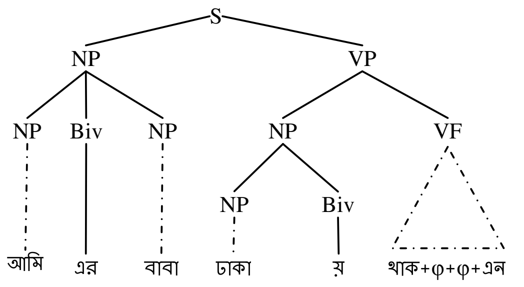

Fig. 1: Syntax tree of an assertive-affirmative sentence.

Derived in one step
Derived in more than one step

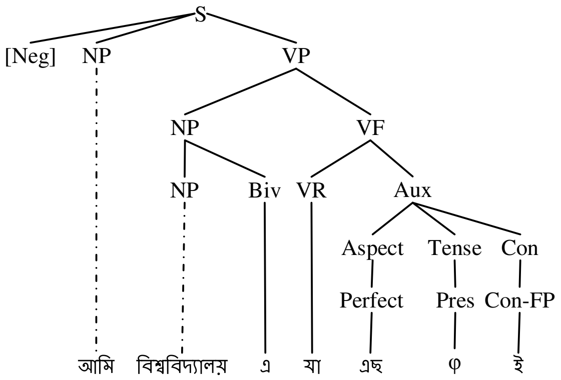

Fig. 2: Syntax tree for আমি বিশ্ববিদ্যালয়ে গিয়েছি

As we can see, if we eliminate the indeclinable word না from a sentence, we get the affirmative forms like আমি সভায় উপস্থিত ছিলাম, আমি বিশ্ববিদ্যালয়ে গিয়েছি, etc. Such assertive-affirmative sentences can be represented by the PSR 1. Therefore, in order to generate the assertive negative sentences we can use the following PSR.

PSR 2: **S → [Neg] + NP + VP**

The abstract symbol [Neg] indicates that the SS of a sentence generated using this rule have negative form which can be obtained by applying a TR 1 as given bellow.

TR 1

> Delete the abstract symbol [Neg]. Check the verb form. If the *tense* and *aspect* [6] of the verb form is [+Present] and [+Perfect] respectively, change the *aspect* to [+Indefinite] and add নি at the end of the verb form. In all other cases, add the word না after the verb form.

Fig. 2 shows the DS of the sentence আমি বিশ্ববিদ্যালয়ে যাইনি. The SS is generated from this DS by applying the *TR 1* as shown in Fig. 3.

Step	**Derivation**	**Rules Applied**
1	**[Neg]** + আমি + বিশ্ববিদ্যালয় + এ + যা + + φ + ই	PSG rules (DS)
2	আমি + বিশ্ববিদ্যালয় + এ + যা + **φ** + φ + ই **+**	TR 1 (SS)
3	আমি বিশ্ববিদ্যালয়ে যাইনি	Morphological rules

Fig. 3: Generation of SS from the DS shown in Fig. 2.

2.2 Interrogative Sentences

We have found the following interrogative words (IWs) shown in Table 1 in Bengali interrogative sentences.

Table 1: Bengali IWs, their types and applications

Parts of Speech	IWs	Example
Pronoun(সর্বনাম) (Considered as noun in language processing)	কে	কে এসেছে? সে তোমার কে? (তুমি) কার সাথে (কে+এর+সাথে) কথা বলছ?
	কোথা	সে কোথা হতে (কোথা+হতে) এসেছে? সে কোথায়(কোথা+য়) গিয়েছে?
	কী	সরকার জনগণকে কী দিয়েছে?
	কোন	(তুমি) কোনটি (কোন+টি) চাও?
	কিসে	সে কিসের জন্য (কিসে+এর+জন্য) এটা করল?
Adverb(ক্রিয়াবিশেষণ)	কিভাবে	সে এটা কিভাবে করল?
	কখন	সে কখন ফিরবে?
Adverb(বিশেষণের বিশেষণ)	কত/কতো	কত হাজার লোক সভায় এসেছিল?
	কয়	কয় হাজার লোক সভায় এসেছিল?
Adjective(বিশেষন)	কী	সে তোমাকে কী জিনিস দিয়েছে?
	কেমন	এটা কেমন জিনিস?
	কত/কতো	কত লোক সভায় এসেছিল? কতজন লোক সভায় এসেছিল?
	কোন	(তুমি) কোন শিক্ষককে পছন্দ কর?
Indeclinable (অব্যয়)	কি	সরকার কি জনগণকে নিরাপত্তা দিয়েছে?
	কেন	সে কেন একাজ করছে?
	কিসে	কিসে এটা হল?

As we see, there are four types of IWs: pronoun, adverb, adjective and indeclinable. The inflections (Bivokti) can be added with the pronoun IWs only, e.g., কার(কে+এর), কোথায়(কোথা+য়), কিসের জন্য (কিসে+এর+জন্য), etc. Some pronoun and adjective IWs can take post positions (অনুসর্গ), e.g., কোনটি(কোন+টি), কতজন(কত+জন). They can be parsed using PSG and TGG rules explained in [6]. As an example, we parse the noun phrase কতজন লোক here bellow using rules described in [6]. Note that the adjective কত is a quantifier.

Step	Derivation	Applied Rule/Comments
1.	#NP#	Starting Symbol
2.	#Spcfr+N#	NP → Spcfr+N
3.	#Qntfr+PP+N#	Spcfr → Qntfr+(PP)
4.	#কত+জন+লোক#	Binding
5.	#কতজন লোক#	Morphological Rule

Let us consider the Bengali interrogative sentence সরকার জনগণকে কী দিয়েছে?. There is little syntactic difference of this sentence with an assertive-affirmative sentence. For example, the assertive-affirmative sentence সরকার জনগণকে নিরাপত্তা দিয়েছে, where the word নিরাপত্তা and কী both are nouns, has the same structure with above sentence, except that the noun নিরাপত্তা is not an IW, while the word কী is a IW and the sentence does not end with a '?' mark.

Now let us consider another interrogative sentence সে কখন ফিরবে? consisting of an adjective IW কখন. The word আজ and কখন are both adjective. The assertive-affirmative sentence সে আজ ফিরবে and the interrogative sentence সে কখন ফিরবে? has the same structure except that the later one ends with a '?' mark and the word কখন is a IW, while আজ is not.

On the other hand, the interrogative sentence সরকার কি জনগণকে নিরাপত্তা দিয়েছে?, consisting of an indeclinable word কি, has different structure than an assertive-affirmative sentence. However, if we eliminate the IW কি and the question mark '?' from the sentence we get সরকার জনগণকে নিরাপত্তা দিয়েছে which is an assertive-affirmative sentence.

Therefore, in the first two cases we see that there is no syntactic difference between an interrogative sentence and an assertive sentence except the last '?' marker in the interrogative sentence. On the otherhand, in the third case we can eliminate the syntactic differences by performing some elimination operations. Therefore, we can define the PSG and TGG rules as follows.

PSR 3: **S → [Int] + NP + VP**

The abstract symbol [Int] activates the following TR:

TR 2

> If the IW has the property [+Adjective] or [+Pronoun], eliminate the abstract symbol [Int] and attach a question sign '?' at the end of the sentence.
> Else if the IW has the property [+Indeclinable], eliminate the abstract symbol [Int], place the IW just after the first NP and attach a question sign '?' at the end of the sentence.

As an example let us generate the sentence সরকার কি জনগণকে নিরাপত্তা দিয়েছে?. The first step is to genetae the DS সরকার জনগণকে নিরাপত্তা দিয়েছে using the PSR 3. The Fig. 4 shows the DS. The grammar used here is described in [6]. The next step is to apply the TR 2 to generate the SS সরকার কি জনগণকে নিরাপত্তা দিয়েছে?. This is shown in Fig. 5.

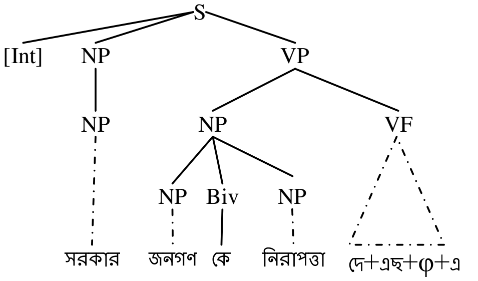

Fig. 4: Syntax tree for সরকার জনগণকে নিরাপত্তা দিয়েছে

Step	Derivation	Rules Applied
1	**[Int]**+সরকার+জনগণ+কে+নিরাপত্তা+দে+এছ+φ +এ	PSG rules
2	সরকার+কি+জনগণ+কে+নিরাপত্তা+দে+এছ+φ+এ+?	TR 2
3	সরকার কি জনগণকে নিরাপত্তা দিয়েছে?	Morphological rules

Fig. 5: Generation of SS সরকার কি জনগণকে নিরাপত্তা দিয়েছে?

2.3 Negative-Interrogative Sentences

The PSR for generating negative-interrogative sentences is as follows:

PSR 4: **S → [Int] + [Neg] + NP + VP**

Here the abstract symbol [Neg] activates the transformation rule TR 1 and [Int] activates the transformation rule TR 2. As an example, let us generate the sentence সরকার কি জনগণকে নিরাপত্তা দেয়নি?. The DS of this sentence is সরকার জনগণকে নিরাপত্তা দিয়েছে as shown in Fig. 4. After generating the DS, [Neg] activates the TR 1 which generates the first level SS সরকার জনগণকে নিরাপত্তা দেয়নি . In the final step [Int] activates the TR 2 and generates the final level SS সরকার কি জনগণকে নিরাপত্তা দেয়নি?

2.4 Imperative Sentences

There is little syntactic difference between assertive sentences and imperative sentences. In imperative sentences, if the subject is second person then the subject may remain hidden. The verbs in such sentences have special form called *imperative mood*. As we have discussed in [6], a verb has two parts: *root* and *auxiliary.* The appearance of the root never changes but the auxiliary part changes based on person and class of the subject and tense of the action. Table 2 shows the auxiliary part of imperative forms of verbs for various tense and person and classes. By inspecting the auxiliary part of a verb we can determine if a sentence is imperative or not.

Table 2: *Verb Auxiliary* part of imperative forms of verb

Person (Class)	**Subject**	**Verb Auxiliary (Example)**		
		Present	**Past**	**Future**
Second and Third Honorific	আপনি/আপনারা/তিনি/তারা	উন (করুন)	*Not applicable*	বেন (করবেন)
Second Non-honorific	তুমি/তোমরা	অ (কর)	*Not applicable*	ও (করো)
Second Pejorative	তুই/তোরা	φ (কর্)	*Not applicable*	স/ইস (করিস)
Third Non-honorific	সে/তারা	উক (করুক)	*Not applicable*	বে (করবে)

We can generate imperative sentences using the following PS rule:

PSR 5: **S → [Imp] + NP + VP**

The abstract symbol [Imp] will activates the following transformation rule:

TR 3

1. Since Aspect of the verb is [+Imperative], when forming the verb, use with the verb-root the verb-auxiliary as shown in Table 2.
2. If the NP is second person, if necessary, eliminate NP. Delete the abstract symbol [Imp].

Now let's generate an imperative sentence বাড়িতে এসো. The Fig. 6 shows the syntax tree, i.e., the DS. After applying TR3 on DS, SS structure বাড়িতে এসো is obtained as shown in Fig. 7.

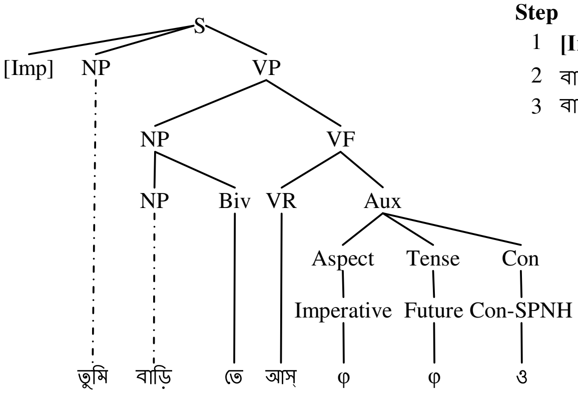

Fig. 6: Syntax tree for তুমি বাড়িতে এসো

Step	Derivation	Rules Applied
1	**[Imp]**+তুমি+বাড়ি+তে+আস্+φ+φ+ও	PSG rules
2	বাড়ি+তে+আস্+φ+φ+ও	TR 3
3	বাড়িতে এসো	Morphological rules

Fig. 7: Generation of SS বাড়িতে এসো

2.5 Negative-Imperative Sentences:

The negative-imperative sentences do not have present form. They are always in future form. For example, in the imperative sentence স্কুলে যা, the verb যা is in present form. However, negative form of this sentence স্কুলে যাস না has the verb যাস (যা+স) in future form as shown in Table 2.

The Negative-interrogative sentences can be represented as:

PSR 6: **S → [Neg] + [Imp] + NP + VP**

Here the abstract symbol [Imp] and [Neg] activates the following transformation rule.

TR 4

1. If tense is [+Present], set tense [+Future].
2. Apply TR 3
3. Apply TR1

Here, the first two rules of TR 4 produce স্কুলে যাস and the third rule generates final SS স্কুলে যাস না.

3. Conversion from SS to DS

Here we present an algorithm to convert the SS of a sentence into it's DS. The reverse process, i.e., converting from the DS to SS, is rather trivial because after finding the DS, based on the type of the abstract symbol found in the DS, transformation rules shown in section 2 are applied to generate the SS.

Like PSRs, TRs are not strictly formal. Therefore, converting from SS to DS is not as trivial as it's reverse process. We devise the following algorithm to convert from the SS of a sentence to its DS. The output of this algorithm can be fed to a bottom up parser to generate the syntax tree.

Algorithm to convert from SS to DS
Input: A Bengali sentence
Output: DS of the sentence

1. Take the whole sentence and breaks it into individual words. Each word is then checked in the lexicon (dictionary) for validity. If necessary more than one words can make a group.

2. Break each of the words or word groups into its constituents called lexemes. For example, the sentence সে বই পড়ছে না is broken into sequence of lexemes (সে+বই+পড়+ছ+φ+এ+না). The lexemes are then tokenized with the help of a dictionary and some rules (to eliminate ambiguity). After tokenization, the above sequence of lexeme looks like (TPNH + TPNH + VR + Continuous + φ + Con-THNH + Indeclinable) [6].
3. Check the lexeme of the last token. If it is a '?', the sentence is an interrogative. If so add [Int] at the front of the lexeme sequence and eliminate last token containing lexeme '?'. Find the IW from the sequence of lexemes. If the IW has the property [+Indeclinable], eliminate it.
4. If the lexeme of the last token is না, add [Neg] at the front of the lexeme sequence and eliminate the last token containing lexeme না. For example, the above sequence will look like ([Neg] + TPNH + TPNH + VR + Continuous + φ + Con-THNH)
5. If the sentence is not interrogative, check if it is imperative or not. Form the auxiliary part of the verb from the lexeme sequence. In the above example, auxiliary will be ছ+φ+এ = ছে. If the verb auxiliary is not in normal mood, check if it is in imperative mood. If so, check if the first noun (e.g., TPNH) agree with the auxiliary or not. If does not agree, add a dummy noun, after the abstract symbol, if any, which agrees with the auxiliary. Add an [Imp] abstract symbol at the front of the token sequence.

The token sequence resulted from the algorithm constitutes the DS of the sentence. In the above example, the DS is ([Neg] + TPNH + TPNH + VR + Continuous + φ + Con-THNH). This DS is now ready to be parsed using some bottom up parsing algorithm. The output of the parser is the syntax tree. Parsing is out of the scope of this paper.

4. Discussion

Use of TGG in natural language processing is not new [2, 3, 4, 6, 12, 13]. However, their use in Bengali natural language processing is new. So far most of the research works on Bengali syntax analysis consider assertive sentences only [5, 7, 8, 9, 10, 11]. No significant work has been done on other type of sentences except the preliminary analysis in our previous work [6].

We consider different cases of assertive, interrogative and imperative sentences in this paper. However, it is difficult to guarantee that we have not missed some cases. Although we have formalized our work using formal grammar and algorithms, we have not implemented our work yet. Moreover, most of the natural language sentences are complex or compound rather than simple. It is still a big issue how our work can be used in such type of sentences.

5. Conclusion

Analyzing and designing machine-processable grammars to understand and generate various types of Bengali sentences is a precondition to develop Bengali natural language applications. This paper paves the way in this direction. In this paper, we have analyzed various types of sentences and designed transformational generative grammar rules for them. Transformation between deep structure and surface structure of various types of sentence is the main focus of this paper. We have limited our discussion only on the mostly used types of sentences, i.e., assertive, interrogative and imperative sentences. We have not considered complex and compound sentences also. Such types of sentences will be considered in our future research work.

References

[1] Sunit Kumar Chatterji, *Bhasha Prakash Bangla Vyakaran* (ভাষা-প্রকাশ বাঙ্গলা ব্যাকরণ), Rupa and Co., Calcutta, 1996.

[2] Humayun Azad, Bakyatattya(বাক্যতত্ত্ব), The University of Dhaka, Dhaka, 1994.

[3] *Transformational grammar*. (2009, June 6). In Wikipedia, The Free Encyclopedia. Retrieved 14:21, June 6, 2009, from http://en.wikipedia.org/w/index.php?title=Transformational_grammar&oldid=294785325.

[4] Viator Lumbanraja, *A Brief Study of Structural-Phrase and Transformational-Generative Grammar: A Syntactic Analysis*, February 2005, Vol. 8 No. 3, www.usd.ac.id/06/publ_dosen/phenomena/83/viator.pdf.

[5] Md. Manzur Murshed, *Parsing of Bangla Natural Language Sentences*, In Proc. of International Computing and Information System, Dhaka, 1998.

[6] Mohammad Reza Selim and Muhammed Zafar Iqbal, *Syntax Analysis of Phrases and Different Types of Sentences in Bangla*, Proceedings of International Conference on Computer and Information Technology, SUST, 3-5 December, 1999, pp. 175-186.

[7] M. M. Hoque and M. M. Ali, *A Parsing Methodology for Bangla Natural Language Sentences*, In Proc. International Conference on Computer and Information Technology, ICCIT'03, Dhaka, Bangladesh, vol. 2, pp. 277-282, 2003.

[8] G. Hossian, A. Kabir and N. Islam, *Development of Smart Bangla Conversion Processor: A New Approach*, In Proc. National Conference on Computer Processing in Bangla, Dhaka, Bangladesh, February 2005.

[9] M. M. Hoque and M. M. Ali, *Semantic Features and Redundancy Rules for Analyzing Bangla Sentences*, In Proc. International Conference on Computer and Information Technology, ICCIT'05, Dhaka, Bangladesh, pp. 1198-1201, vol. 4, 2005

[10] Naira Khan and Mumit Khan, *Developing a Computational Grammar for Bengali using the HPSG Formalism*, In Proc. of 9th International Conference on Computer and Information Technology (ICCIT 2006), Dhaka, Bangladesh, December 2006.

[11] Md. Nasimul Haque and M. Khan, *Parsing Bangla using LFG: An Introduction*, BRAC University Journal, Vol 2, No. 2, 2005.

[12]Abul Kalam Manzur Morshed, *Adhunik Bhasatatto (আধুনিক ভাষাতত্ত্ব)*, Naya Udyog, Calcutta, 1997.

[13] Humayun Azad, *Pronominalization in Bengali*, The University of Dhaka, Dhaka, 1983.

Submitted: 16th August, 2009; **Accepted for Publication:** 15th November, 2009.