Squibs and Discussions

A Delayed Syntactic-Encoding-based LFG Parsing Strategy for an Indian Language—Bangla

Probal Sengupta*
Indian Statistical Institute

B. B. Chaudhuri*
Indian Statistical Institute

1. Introduction

In this squib, we propose a technique aimed at efficient computer implementation of LFG-based parsers for Indian languages in general and Bangla (Bengali) in particular. (For the LFG formalism, see Kaplan and Bresnan [1982].) The technique may also be useful for other languages having similar properties.

Indian languages are mostly nonconfigurational and highly inflectional. Grammatical functions (GF's) are predicted by case inflections (markers) on the head nouns of noun phrases (NPs) and postpositional particles in postpositional phrases (PPs). However, in many cases the mapping from case marker to GF is not one-to-one. The classical technique for non-configurational syntactic encoding of GF's (Bresnan 1982b) therefore requires a number of alternations to be thrown in to handle this phenomenon. The resulting nondeterminism in the parser implementation leads to a non-efficient unification component. The problem here, however, is not of unbounded functional uncertainty (described, with proposed solutions, in Kaplan, Maxwell, and Zaenen [1987], Kaplan and Maxwell [1988], and Kaplan and Zaenan [1990]), but rather, one of disjunctive constraint satisfaction bounded within the matrix. Disjunctive constraint satisfaction leads to a degradation of efficiency of the unification component of LFG, as has been pointed out in Knight (1989) and Maxwell and Kaplan (1991).[1] A closer look at the languages reveals that most disjunctions do not exist if an a priori knowledge of the verb (which is generally at the end of the sentence, since Indian languages are mostly verb final and the verb is the last lexeme encountered in a left-to-right scan of the parser) is available. Here we propose a technique that uses this fact to reduce alternations in syntactic encoding. Our method is based on a delayed evaluation of syntactic encoding schema. We treat the points of syntactic encoding of noun phrases as **forward references** that are temporarily maintained in a symbol table for later binding. A new metavariable, augmentation of the scope of the Locate operator, and a special type of schema (called **m-structure**) to be projected by the verb are some of the salient features of our technique.

2. Delayed Syntactic Encoding

As suggested in Bresnan (1982a, 1982b) and Mohanan (1982), a flat constituent structure for a Bangla sentence S is given by the rule in (1), where constituent NPs and/or

Table 1
Case markers and their possible grammatical functions. The GEN case marker normally marks a genitive qualifier of a noun. However, for certain verb forms (for example, ones in pseudopassive voice), it also marks the subject.

Marker	Name	SUBJ	OBJ	IOBJ	ADJUNCT
None	NULL	•	•	•	•
+ke, +re	DATive	•	•	•	
+e, +te	OBLique	•	•	•	•
+er	GENitive	•			

PPs may freely permute among themselves.[2]

$$S \longrightarrow \underset{(\uparrow\ (\downarrow\ CASE)) = \downarrow}{NP^*} \quad V \quad \underset{(\uparrow\ (\downarrow\ CASE)) = \downarrow}{NP^*} \tag{1}$$

$$(\uparrow\ (\downarrow\ CASE)) = \downarrow \tag{2}$$

In (1), syntactic encoding of GF's is carried out using the simplified encoding schemata (2) annotating the NPs (Bresnan 1982b, 297–299). In the implementation domain, schemata (2) works quite well if the mapping from case marker to function is nearly one-to-one. Unfortunately, as shown in Table 1, many modern Indian languages lack this property—almost every marker has many-to-one mapping. The classical way of handling such situations is to use alternation or disjunction. However, in the context of an a priori lexical knowledge of the verb, the alternations cease to exist in most cases. To express this more formally, let $G = \{g_1, g_2, \ldots\}$ be the set of relevant GF's, $C = \{c_1, c_2, \ldots\}$ be the set of NP case markers, and *cToG* be a mapping from case markers to GF's, such that $cToG(c)$, $c \in C$ is (are) the grammatical function(s) predictable from c. In our case, $cToG(c)$ is actually a finite disjunction $g_{i_1} \vee g_{i_2} \vee \cdots$ of functions. If f_N is the f-structure of an NP of a sentence S with f-structure f_S and the case marker on the head noun of the NP is c, the semantics of schemata (2) annotating the NP is $(f_S\ cToG[c]) = f_N$, where "=" denotes unification. Since $cToG[c]$ is a disjunction, in a parser implementation, it effectively multiplies out to $|cToG(c)|$ *nondeterministic* choices for the functional role played by the f_N in f_S. If, in the ultimate analysis, the NP is found to play the functional role g in f_S, the constraints set in (3), projected by the verb, must have been satisfied:

$$\begin{array}{ll} & (f_S\ g\ CASE) = c \\ \wedge_i & (f_S\ g\ q_i) = v_i \end{array} \tag{3}$$

where q_i are different normal agreement features (like NUMBer, PERSon, etc.) and/or other semantic agreement features (like ANIMacy, etc.). We shall call the schema (3) the **agreement schema** for the function g projected by the verb. Observations show that in most well-formed sentences, the agreement schema of the verb for any function g is satisfied by *at most one* constituent NP of the sentences, provided some order of processing the agreement schema of different GF's is maintained. The mapping *cToG* is therefore nearly one-to-one in the context of the agreement schema of the verb and the

2 The PPs have been kept out of the present discussion.

agreement schema may serve as test criteria for selecting grammatical functions from internal properties of NPs. The parser must ensure evaluation of an encoding schemata of a constituent NP in the context of the agreement schema of the verb, somewhat like handling a forward reference (where an item referred to is defined later than the places where it has been referred to). The trick is to delay the evaluation of encoding schema of constituent NPs till an appropriate moment, while maintaining a persistent data structure, such as a symbol table, to keep track of the points of forward reference (at which actual function names get instantiated) and their local environments (the internal f-structure of the constituent NPs).

3. The Proposed Solution Technique

In this section, we provide the basic solution technique for simple sentences (i.e., consisting of a single verb only) in two parts.

3.1 Solution Part I: Initiation of Forward Reference

A forward reference discussed in the previous section is encountered during Locate-ing the left-hand side of a schemata like (2) while processing an NP. In our delayed encoding proposal, the (modified) Locate operation should leave the "name" of the functional role played by the NP as "underspecified." To force the Locate operator to behave in this manner, we propose:

1. The introduction of a new type of underspecification metavariable: ?
2. The modification of encoding schemata (2) to schemata (4):

$$(\uparrow ?) = \downarrow \tag{4}$$

The ? metavariables generate placeholders for hitherto anonymous grammatical functions, which we shall call **nameholders**, and denote them by actual name variables $n_1, n_2, \ldots$. Locate-ing of schemata (4) creates such a nameholder (n, say) in the scope of the functional placeholder (f, say) for the $\uparrow$ metavariable and simultaneously stores the pair (f, n) in the symbol table. Locate-ing a construct like $(f\ n)$ where both f and n are already defined placeholder and nameholder, respectively, returns (a pointer to) the "value" part of the pair in the f-structure (pointed at by) f, whose name is (pointed at by) n. The extended semantics of Locate is therefore:

Locate[d], where d has the form $(x\ y)$. Let f be the reference to an f-structure Locate[x]. If y is a ? metavariable, let n be a new nameholder for the metavariable. An anonymous slot is created in the scope of f, and n is made to point to it. Simultaneously, the pair (f, n) is entered as a new entry of the symbol table. If, however, y is a nameholder n, Locate returns the value field of the pair in f whose name field is held by n.

With this, the semantics of Locate with respect to the form in (5), which is the left-hand side of schemata (4), may be pictorially represented as in Figure 1.

$$(\uparrow ?) \tag{5}$$

3.2 Solution Part II: Name Binding of Forward References

The next point to be considered is binding actual function names to nameholders. We assume that the agreement schema for a function g may select the structure that satisfies the constraints. For this, the agreement schema must be handled in a different

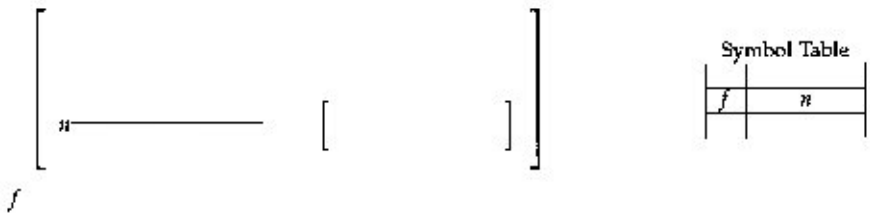


Figure 1
Semantics of Locate with respect to (5).

manner than normal projection schema. We choose the notation $(\#\ g\ q_1) = v_1$ for one agreement schemata for the function g. We shall call the forms $(\#\ g\ q_i) = v_i$ a metastructure or m-structure. M-structure schema are projected by the main verb of a sentence. A symbol table entry (f, n) satisfies an m-structure schemata $(\#\ g\ q_i) = v_i$ projected by the verb V of a sentence S, if f is the f-structure of S, and the structure $(f\ n)$, where n is treated as an atom, contains the pair $[q_i\ \ v_i]$. If a symbol table entry satisfies all m-structure schema for a function g, by our proposed scheme, the nameholder n that points to the entry is bound to the function name g. Also, the satisfying symbol table entry is deleted.

Testing of symbol table entries with m-structure schema and resulting binding of namcholders to actual function names are carried out by a newly introduced operator Search. The operator Search takes the entire set m-structure schema for a particular CF and carries out the process described in the previous paragraph. If more than one symbol table entry satisfies the m-structure schema for a particular function g, the one earlier in order of occurrence is chosen. The relative evaluation (by operating with Search) order for the sets of m-structure schema for different functions is motivated by the default ordering of phrases in a sentence in the target language. In Bangla for example, the default ordering is SUBJ-IOBJ-OBJ. Thus, the test for SUBJ is carried out first, followed by IOBJ, and OBJ, if any.

The final solution technique therefore involves first evaluating all f-structure schema, including those with underspecification metavariables annotating the children nodes of an S-dominated c-structure tree. This would generate symbol table entries corresponding to NPs annotated with the ? schema. Next, the m-structure schema of the main verb are operated on with the Search operator in the default phrasal order for the language. A sentence is well formed if and only if all the m-structure schema for the verb are satisfied and all nameholders in the scope of the sentence are bound to names (i.e., at the end, the symbol table is empty). The evaluation process naturally satisfies the uniqueness property for sentence-level grammatical functions.

Regarding the relative evaluation order of f- and m-structure schema, the general principle is "all f-structure schema are evaluated before any m-structure schemata is evaluated (i.e., fed to the Search operator)."

Example 1
Let us consider the Bangla simple sentence below, in which the NPs have been underlined.

a'pni a'ma'ke ekt'a' bai dilen
You(honored)-NULL I-DAT one-DEF book-NULL give-3p-hon-PAST
You (honored) will give me a book

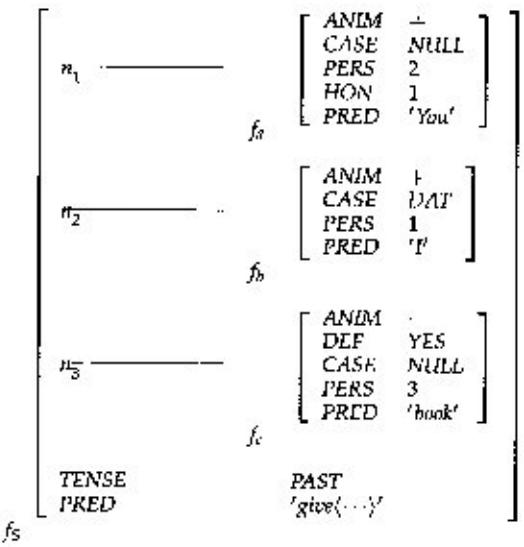
a) **a'pni** *N* (↑ *PERS*) = 3,
(↑ *HON*) = 1,
(↑ *ANIM*) = +,
(↑ *PRED*) = 'you',
(↑ *CASE*) = *NULL*

b) **a'ma'ke** *N* (↑ *PERS*) = 1,
(↑ *HON*) = 0,
(↑ *ANIM*) = +
(↑ *PRED*) = 'I',
(↑ *CASE*) = *DAT*

c) **bai** *N* (↑ *PERS*) = 0,
(↑ *HON*) = 0,
(↑ *ANIM*) = -,
(↑ *PRED*) = 'book',
(↑ *CASE*) = *NULL*

d) **dilen** *V* (↑ *TENSE*) = *PAST*
(↑ *PERS*) = 2
(↑ *HON*) = 1
(↑ *PRED*) = 'give((*SUBJ*), (*OBJ*), (*IOBJ*))'
(# *SUBJ PERS*) = 2
(# *SUBJ HON*) = 1
(# *SUBJ ANIM*) = +
(# *SUBJ CASE*) = *NULL*
(# *IOBJ ANIM*) = +
(# *IOBJ CASE*) = *DAT*
(# *OBJ ANIM*) = -
(# *OBJ CASE*) = *NULL*

Figure 2
Lexical entries of head nouns and verbs in Bangla sentence *a'pni a'make ekt'a' bai dilen*.



Symbol Table

f_S	n_1
f_S	n_2
f_S	n_3

Figure 3
F-structure of Bangla sentence *a'pni a'ma'ke ekt'a' bai dilen*.

Any permutation of the underlined phrases and the verb should give identical results. The lexical entries of the head nouns and the verb are given in Figure 2. The feature HON is a three-valued scalar, 1 for honored, 0 for casual, and -1 for intimate. Since Bangla has no subject-verb agreement based on number, the NUM feature has been omitted.[3]

The f-structure f_S of the sentence before processing the m-structure of the verb appears as in Figure 3 and the final solution is as given in Figure 4. The f-structures f_a, f_b, and f_c are for the NPs in order.

3 Alternately, since Bangla verbs are not marked for number, the NUM feature is omitted in agreement.

$$
\left[\begin{array}{ll}
SUBJ & \left[\begin{array}{ll} ANIM & + \\ CASE & NULL \\ PERS & 2 \\ NUM & 1 \\ PRED & \text{'You'} \end{array}\right] \\
IOBJ & \left[\begin{array}{ll} ANIM & + \\ CASE & DAT \\ PERS & 1 \\ PRED & \text{'I'} \end{array}\right] \\
OBJ & \left[\begin{array}{ll} ANIM & - \\ DEF & YES \\ CASE & NULL \\ PERS & 3 \\ PRED & \text{'book'} \end{array}\right] \\
TENSE & PAST \\
PRED & \text{'give}\langle(SUBJ),(OBJ)(IOBJ)\rangle\text{'}
\end{array}\right]_{fs}
$$

Figure 4
Final solution for Bangla sentence *a'pni a'mia'ke ekt'a' bai dilen.*

4. Discussion

A comparison of our technique and other feature-based parsing mechanisms may be in order. We provide brief comparisons with two such formalisms. In Rambow (1994), a V-TAG parser (a Tree Adjoint Grammar extended to handle scrambling and other aspects) is implemented through the {}-BEPDA,[4] which uses sets of auxiliary trees as unfulfilled nominal subcategorization. More recently, Johnson and Dorre (1995) have presented a framework for constraint coroutining to deal with linguistic constraints that cannot be effectively resolved during parsing at the location in which they are most naturally introduced.

A properly designed NLP platform for Indian languages must come with an efficient morphosyntactic unit for parsing words into their constituent morphemes where lexical projections of words can be obtained from projections of individual morphemes. At present, we have a fully implemented morphosyntactic lexical subsystem for Bangla based on a formalism suitable for an LFG-based NLP platform, as proposed in Sengupta and Chaudhuri (1993) and Sengupta (1994). We have implemented the operators `Locate` (modified as suggested in the text), `Merge` (as an object-oriented unification method), `Include`, and `Search`, and are in the process of creating an effective object-oriented parser for c-structure generation.

We tested our formalism on a sample of about 250 simple and complex sentences picked from newspaper clippings. Though phrasal orderings were quite random, almost all simple sentences in active voice (constituting about 57% of the samples) were correctly parsed. The method has been extended to take care of a class of complex sentences (with dependent clause "embedded" within the matrix) and "chained" clauses as described in the results (not LFG based) in Sengupta (1994).

4 A multiset version of Bottom-up Embedded Push-down Automata.

References

Bresnan, J. 1982a. Passive in lexical theory. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA, pages 3–86.

Bresnan, J. 1982b. Control and Complementation. In J. Bresnan, editors, *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA, pages 282–390.

Johnson, M. and J. Dorre. 1995. Memorization of Coroutined Constraints. In *Proceedings of the 33rd Annual Meeting*, pages 100–107, Cambridge. Association for Computational Linguistics.

Kaplan, R. M. and J. Bresnan. 1982. Lexical Functional Grammar: A Formal System for Grammatical Representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA, pages 173–281.

Kaplan, R. M., J. T. Maxwell, and A. Zaenen. 1987. Functional Uncertainty. In *The CSLI Monthly*, Center for the Study of Language and Information, Stanford University.

Kaplan, R. M. and J. Maxwell. 1988. An Algorithm for Functional Uncertainty. Technical Paper No. P88-00084, Systems Sciences Laboratory, XEROX PARC. (Also in *Proceedings of COLING-88*, Budapest, pages 297–302.)

Kaplan, R. M. and A. Zaenen. 1990. Long-distance Dependencies, Constraint Structure and Functional Uncertainty In M. Baltin and A. Kroch, editors, *Alternate Conceptions of Phrase Structure*. Chicago University Press, Chicago.

Knight, K. 1989. Unification: A Multidisciplinary Survey. *ACM Computing Surveys* 21(1):93–124.

Maxwell, J. T. and R. M. Kaplan. 1991. A Method for Disjunctive Constraint Satisfaction. In M. Tomita, editor, *Current Issues in Parsing Technology*. Kluwer Academic Publishers, pages 173–190.

Mohanan, K. P. 1982. Grammatical Relations and Clause Structure in Malayalam. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge, MA, pages 504–589.

Rambow, O. 1994. *Formal and Computational Aspects of Natural Language Syntax*. Ph.D. dissertation, University of Pennsylvania.

Sengupta, P. and B. B. Chaudhuri. 1993. A Morpho-Syntactic Analysis Based Lexical Sub-System. *International Journal of Pattern Recognition and Artificial Intelligence* 7(3):595–619.

Sengupta, P. 1994. *On Lexical and Syntactic Processing of Bangla Language by Computer*. Ph.D. thesis, Indian Statistical Institute.