

Towards Convenient Bi-Directional Grammar Formalisms

P. Newman
IBM Palo Alto Scientific Center
1530 Page Mill Road
Palo Alto, CA 94304, USA

Abstract

This paper discusses the advantages for practical bi-directional grammars of combining a lexical focus with the GPSG-originated principle of immediate-dominance/linear-precedence (ID/LP) rule partitioning. It also outlines an implementation approach following these guidelines. The approach is inspired by Slot Grammar, with additions including more explicit mappings between surface and internal representations, and preferential constituent ordering rules.

1. Introduction

The term *bi-directional grammar formalism* here refers to an implementation formalism capable of producing grammars usable in both analysis and generation. Such grammars can be advantageous for machine translation and other applications for reasons of economy; they also aid in grammar validation, as suggested by Dymetman and Isabelle (1988).

There have been major strides taken in recent years in bi-directional formalisms, based on many different paradigms. In many cases some elements of the specifications are direction-unique. However, bi-directionality is not an end in itself, as contrasted with its potential gains. So the goal can be usefully approximated with formalisms which make some limited distinctions between information applying to parsing and to generation.

Among recent efforts in this area are: (a) the CRITTER system described by Dymetman and Isabelle (1988), in which an annotated definite clause grammar is compiled differently, based on the annotations, for the two purposes, (b) the inversion of a systemic generator by Kasper (1988) (in which phrase structure is said to be added manually for parsing), (c) the DEW generator of Caraceni and Stock (1988), which is based on an augmented transition network (ATN) and which seems to employ a "generate and test" approach to generation, and (d) the Berlin GPSG effort of Buseman and Hauenschild (1988) in which GPSG (Gazdar et al. 1985) is adapted for implementation purposes to allow feasible rule specification and sequencing.

The purpose of this paper is to suggest that for future work in operational bi-directional formalisms, approaches combining a high degree of lexicalism with some form of GPSG-inspired ID/LP partitioning of information appear especially promising. Some formalisms with these characteristics are the head-driven grammars HPSG (Pollard and Sag 1987), and Slot Grammar (McCord 1989a). The latter is currently used in the machine translation system LMT-2 (McCord 1989b), but for parsing only. Aspects of Slot Grammar

will be used to illustrate the discussion, which is in four parts. Section 2 discusses the relationship of a strong lexical component to bi-directionality. Section 3 discusses the difficulties of obtaining realistic bi-directional grammars without an ID/LP separation. Section 4 discusses ways in which head driven grammars, in particular, Slot Grammar, avoid these difficulties. Finally, section 5 discusses some proposed extensions of Slot Grammar to illustrate a possible organization of information for a head-driven bi-directional grammar.

2. Lexicalism and Bi-Directionality

Probably the majority of contemporary grammars place considerable information in the lexicon. This is especially important in a bi-directional context because it allows direction-free statements of:

- Semantic representations of concepts and their associated modifiers
- Alternative ways in which those semantic representations can be realized in terms of alternative expression of dependents on a syntactic level. This includes identification of pro-forma elements such as required prepositions ("wait FOR John"), and entire pro-forma complements, such as "a hand" in "give a hand", and fixed position information for non-compositional and frozen compounds.
- Unification-oriented mappings between the semantic and syntactic representations.
- Lexical transformations originating in LFG (Bresnan 1982) for changes in lexical form (passivization) and category (e.g., nominalization), as well as alternative realizations of some concepts, for example, those allowed via "raising" and "clefting".

Lexicons incorporating various combinations of these features are used in most of the bi-directional systems mentioned in Section 1. Also, Lancel et al. (1988) claim that significant bi-directionality is obtained in the SAGE system solely through the use of such lexicons, with different syntactic components used in the two processing directions.

3. Inverting Non-ID/LP Grammars

Obtaining realistic bi-directional formulations in grammars not having an ID/LP partitioning of information is problematical because extending their ordering provisions to deal not only with syntactic correctness but also with semantic and textual factors exacerbates an already difficult situation with regard to ordering in such grammars.

To justify this statement, we look first at current trends in non-ID/LP grammars, and then at necessary extensions.

3.1 Trends in Non-ID/LP Grammars

In some contemporary versions of paradigms whose basic rules were originally intended to subsume both "ID" and "LP" information, information is reorganized so the information expressed by the basic rules is quite limited, and additions are needed to express the remainder. The fundamental cause of the modifications is the need to effectively accommodate the relatively free orderings in clause constituents of many languages. Thus, for example, in augmented phrase structure grammars (APSGs), which include definite clause grammars, it is inconvenient to specify each legal dependent ordering by a separate phrase structure rule. Instead, as discussed by Jensen (1987), it is more convenient to focus on binary rules, combining a node containing a head with one of its modifiers, e.g.,

VP0 -> VP1 NP; VP0 -> VP1 PP; etc.

Using these binary forms, the phrase structure portion of the rules indicate not much more than the side of a head on which a modifier may occur. The "augmentations" have a number of responsibilities. They must indicate ordering constraints among siblings in terms of features recording subtree "states" (i.e., to the extent that such states are not implied by the category names). Also, explicit facilities are needed for structure building to avoid separate nodes for each rule applied.¹ Finally, if the rules are used in combination with a lexical orientation, since modifiers are indicated in phrase structure rules by general category (e.g., NP), augmentations must locate and specify the relationship between the modifier category and the specific modifier expressed type (e.g., an expected complement).

The result of this (necessary) movement of function away from the basic rules of the paradigm is a tendency toward somewhat laborious, redundant specification. To illustrate the kind of redundancy involved, we construct an APSG-style binary rule for attachment of indirect objects.

The example assumes a lexicon identifying potential modifiers of a head by frames, which, by unification, map between syntactic and semantic representations of those modifiers. To allow for ordering provisions, modifier frames have associated labels, such as "indobj". We also assume that semantic representations of dependents include valency numbers for complements. To avoid the development of two examples, we also ensure that the rule is bi-directionally applicable.

```
VP0 -> VP1 NP
choosemodifier (VP0,VP1,NP, indobj, M, M1)
unify (VP0,VP1, (hasobj,modifiers))
unify (VP1.hasobj, "-")
union (M,M1,N)
unify (VP0.modifiers, N)
unify (VP1.modifiers, M1);
```

This rule can be understood bi-directionally, if we assume that both the interpreter and "choosemodifier" are direction-sensitive. In parsing the interpreter finds a constituent unifying with (cat = vp) adjacent to one unifying with (cat = np) and instantiates an almost empty constituent VP0 (cat = vp). "Choosemodifier" then checks the lexicon to see if the head of VP1 expects a modifier with label "indobj" whose syntactic subframe unifies with the constituent NP, and whose valency number is not yet found as a modifier in VP1. If so, it returns the result of the full unification as M, and the current value of VP1.modifiers as M1.

"Eunify" destructively unifies two structures except for the listed attributes. This serves in parsing to project head features upward. In parsing the next "unify" function ensures that a direct object has not yet been included in VP1. The remainder of the rule, in parsing, creates the dependent list for VP0 by expanding that of VP1 to include the indirect object.

Similar rules could be constructed for less strictly ordered complements, and for adjuncts.

These operations have considerable inherent redundancy even though much function is abstracted out within "choosemodifier". The "choosemodifier" operation occurs in all complement attachment rules. Feature projection and structure building occur in all rules. Finally, the actual precedence rule aspects can be expressed more perspicuously than via feature state testing. Looking ahead slightly, one way of summarizing the situation is to say that when a grammar paradigm which originally combines "ID" and "LP" resorts to binary rules, especially in the presence of a lexical focus, the grammar becomes, to a large extent, a head-driven grammar, without the ability to take full advantage of the factoring opportunities afforded.

3.2 Extensions for Bi-Directionality

So far we have covered somewhat old ground. Why are these modified approaches especially problematic in a bi-directional context? Because there one is faced with an unpleasant choice between probably untenable complexity and unnecessary generation.

To justify this claim, we return to the rule illustrated in section 3.1, first examining its assumed operation in a generative direction.

In generation the interpreter instantiates almost empty constituents VP1 and NP. "Choosemodifier" then attempts to find an expected modifier frame for the head of VP0 with the given label whose semantic subframe (containing a valency number) unifies with one of the actual modifiers (VP0.modifiers) of VP0, and whose syntactic subframe unifies with NP, returns the result as M, and the remainder of the modifiers as M1, etc.

¹ Similar tendencies are observed in contemporary categorial grammars. For example, Yoo and Lee (1988) use "quotient" categories which specify unordered sets of possible arguments, together with separate LP rules. Bes and Gardent (1989) also use sets within categories, together with order features to constrain adjacency.

But such rules do not really satisfy the requirements of generation. They describe syntactically correct structures but specify no ordering constraints and preferences relating to either semantic considerations (e.g., required orderings of adjective types in English, and conventional orderings of verb modifiers) or to textual considerations such as topic and focus. In parsing such provisions are needed to detect textual features, and in generation they are needed to use textual features to determine ordering. If these provisions were added, the "feature testing" aspects relating to ordering would become considerably more complicated, if expressible at all. This is because detecting and using textual considerations seems to involve taking into account the entire complex of modifiers for a head, which is extremely awkward in terms of binary phrase structure rules.

Hajicova (1989) describes topic/focus determination conditions for both English and Czech; they involve both semantic role information and complex sibling relationships. If those conditions were expressed in the context of binary rules, it seems that a rule such as

VP0 → VP1 x

in the parse would have the responsibility of assigning *x* to "focus" if there has been a break to the left of *x* in the conventional ordering of dependent roles (for Czech), and indeterminate otherwise (until further dependents are found). In generation the rule might be licensed at a stage in generation where *x* is either (a) part of the topic, and VP0 contains only topic dependents, or (b) part of the focus and ranks highest of the dependents in VP0 in the systemic order. (Topic/focus identification criteria for English are also considered by Hajicova, and are more complicated).

So adding textual provisions to phrase structure rules would pose a considerable challenge. Simply put, attaching dependents to heads one at a time is a convenient approach in parsing, but detaching them one at a time is not a convenient approach for generation.

On the other hand, if textual provisions are omitted from the grammar, then generation would produce *all* syntactically legitimate sentences. One would then use additional rule sets to select among all the generated utterances based on semantic and textually based preferences.² (And rules are also needed to detect textual features during analysis.)

4. Head Driven Grammars and Slot Grammar

Head driven grammars which combine a lexical focus with a strict ID/LP partitioning avoid the problems described above. We use Slot Grammar as an example.

The lexicon formulation of Slot Grammar is interesting in that it identifies dependents, both complements and adjuncts, by "slotnames", a device originating in earlier work by McCord (1980). The (alternative) structures which can be used to realize those slots are factored out into separate "filler rules". These rules contain conditions on both prospective fillers and associated heads. They can thus be used to constrain/adjust features of the constituents under consideration, e.g., to instantiate agreement. In other words, they can be used to express many ID constraints.

The basic linear precedence conditions of Slot Grammar are expressed by two types of rules. "Head/Slot" rules indicate the sides of the head on which a particular "slot" may appear. These rules are conditional in terms of unifiers for both head and slot filler. "Slot/slot" rules indicate, again conditionally, precedence rules among slots on the same side of a head.

Organizing information in this way allows the elimination of the explicit specification of many aspects of the rule shown in section 3.1:

1. Association of "categories" with complements and adjuncts is eliminated - ordering is stated in terms of slots rather than the more general syntactic categories.
2. "Choosemodifier" becomes the basic, built-in control operation of the parser, and need not be expressed explicitly
3. Structure building operations are, to a large extent, implicit. Only variations in feature projection, etc need be expressed explicitly.

The remainder of the information in the rule is expressed by two short rules, one which indicates that indirect objects fall on the right sides of heads, and the other that they precede direct objects.

The revised organization of information also provides the basis for dealing with semantic and textually conditioned ordering requirements without either undue complexity or exhaustive generation. This is because the inherent modularity allows the use of different control schemes for parsing and generation. In parsing the control scheme can be "attach one dependent at a time", using immediate dominance rules and basic linear precedence constraints together. In generation the control scheme can be altered to first generate sets of dependents, using just immediate dominance rules, and then

² There have been efforts to combine textual considerations with non-ID/LP grammars. For example, Uszkoreit (1988) uses exhaustive enumeration of alternative modifier orderings, including complements and adjuncts, with selection among alternatives made by a focus feature. However, in parsing, the suggestion must somehow "collapse" to a set-oriented approach, using the enumerated alternatives as a kind of LP rule. Also, as implied by the results of Hajicova (1989), and explicitly argued by Hauenschild (1988), these provisions are not sufficient.

ordering them using both basic precedence constraints and preference-oriented ones.

5. Current Direction

Recapitulating, bi-directional grammar efforts combining lexicalism with an ID/LP separation seem most promising because

1. A lexical focus in itself provides a great deal of bi-directional facility.
2. In current grammars based on non-ID/LP paradigms, linear precedence constraints governing syntactic correctness are expressed by constraints on features of one node of a binary rule. Extending this approach to deal with preferential ordering is at best extremely complex, and possibly infeasible, but the alternative seems to be exhaustive generation followed by filtering.
3. In contrast, the modularity provided by an ID/LP separation allows rules to be applied in different combinations in parsing and generation.

In the work underlying this paper, a multi-lingual machine translation project, a bi-directional grammar formalism is being developed inspired by Slot Grammar, but with modifications including:

1. adding a fully reversible morphological component in the lexicon
2. expanding the lexical provisions to include explicit bi-directional mappings between syntactic and more abstract representations
3. revising the notation to facilitate reversibility.
4. using a slotname type-lattice to simplify the expression of generalizations.
5. adding preferential precedence rules dealing with semantic and textual considerations. The preferential ordering rules are used in the analysis phase to detect textual features, and are applied after a post-parse disambiguation analysis (based on a heuristic search algorithm described in (Newman 1988)). In generation, however, the preferential ordering rules are applied together with those expressing absolute ordering constraints.

A preliminary description of these provisions has been documented (Newman, to appear). The preferential precedence rules are of two kinds: one kind relates to the association of dependents with "zones" of a constituent (e.g., pre-subject, pre-finite,...), and the other kind deals with their ordering within zones. Zones are used because some aspects of dependent ordering are most conveniently described in those terms, as discussed by Quirk et al (1972), Uszkoreit (1988) and others.³ Zone association rules express the preferences of certain types of modifiers for certain zones, and also variations in these preferences due to textual considerations. These

preferences must be balanced, by heuristics, against the needs of other modifiers and the constraints imposed by the zones themselves. Optimal ways of stating and using these preferences represents a major focus of our current work.

6. References.

1. Bes, G.G., Gardent, C. "French Order Without Order", *Proc. 4th Conf. of European Chapter of ACL* (1989), 249-255
2. Buseman S., Hauenschild C. "A Constructive View of GPSG or How to Make It Work", *Proc COLING 88*, 77-82
3. Caraceni, R., O. Stock, "Reversing a Lexically Based Parser for Generation," *Applied Artificial Intelligence*, vol. 2, # 2 (1988) 149-74
4. Dymetman M., Isabelle, P. "Reversible Logic Grammars for Machine Translation", *Proc 2nd Int'l Conf on Theoretical and Methodological Issues in the Machine Translation of Natural Languages* (1988)
5. Gazdar, G., E. Klein, G. Pullum, I. Sag., *Generalized Phrase Structure Grammar*, Basil Blackwell (1985)
6. Hajicova E. "A Dependency-Based Parser for Topic and Focus", *Proc. Int'l Workshop on Parsing Technologies* (1989) 448-457
7. Hauenschild C. "GPSG and German Word Order", in U.Reyle, C. Rohrer, eds., *Natural Language Parsing and Linguistic Theories*, Reidel (1988), 411-431
8. Jensen, K. "Binary Rules and Non-binary Trees", in A. Manaster-Ramer (ed.), *Mathematics of Language*, John Benjamins (1987)
9. Kasper, R. T. "An Experimental Parser for Systemic Grammars", *Proc COLING 88*, 309-312
10. Karttunen L., "Radical Lexicalism", CSLI Report CSLI-86-68 (1986)
11. Lancel, J-M, Otani M., Simonin N., Danlos L. "SAGE: A Sentence Parsing and Generation System", *Proc COLING 88*, 359-364
12. McCord, M.C., "Slot Grammars", *Computational Linguistics*, vol 6, 31-43 (1980)
13. McCord, M.C. "A New Version of Slot Grammar", IBM Research Report RC 14506 (1989a)
14. McCord, M.C. "A New Version of the Machine Translation System LMT", IBM Research Report RC 14710 (1989b), to appear in *Proc. International Scientific Symposium on Natural Language and Logic*, Springer Lecture Notes in Computer Science
15. Newman, P. "Combinatorial Disambiguation", *Proc. ACL Conf. on Applied NLP* (1988)
16. Newman, P. "Symmetric Slot Grammar", to appear in *Proc 3rd Int'l Conf on Theoretical and Methodological Issues in Machine Translation of Natural Languages* June 1990

³ It might be noted that to allow meaningful use of zones, the syntactic structures used in the design are very flat. Fronting is not viewed, as in most current approaches, as an example of long-distance dependency. Rather, to simplify the statement of zone-allocation and other ordering rules, dependents assigned to different zones are siblings and, as in Karttunen (1986), auxiliaries are adjunct-like.

17. Pollard, C. and I. Sag, *Information-based Syntax and Semantics* Vol. 1, CSLI (1987)
18. Quirk, R., S. Greenbaum, G. Leech, J. Svartnik, *A Grammar of Contemporary English* Longman (1972)
19. Uszkoreit, H. *Linear Precedence in Discontinuous Constituents: Complex Fronting in German*, CSLI Research Report CSLI-86-47 (1988).
20. Yoo S., Lee, K. "Extended Categorical Grammar", CSLI Report CSLI-88-121 (1988)