CONSTRAINING A DETERMINISTIC PARSER

J. Bachenko, D. Hindle, and E. Fitzpatrick

Computer Science and Systems Branch
Information Technology Division
Naval Research Laboratory
Washington, D.C. 20375

ABSTRACT

At the Naval Research Laboratory, we are building a deterministic parser, based on principles proposed by Marcus, that can be used in interpreting military message narrative. A central goal of our project is to make the parser useful for real-time applications by constraining the parser's actions and so enhancing its efficiency. In this paper, we propose that a parser can determine the correct structures for English without looking past the "left corner" of a constituent, i.e. the leftmost element of the constituent along with its lexical category (e.g. N, V, Adj). We show that this Left Corner Constraint, which has been built into our parser, leads quite naturally to a description of verb complements in English that is consistent with the findings of recent linguistic theory, in particular, Chomsky's government and binding (GB) framework.

I INTRODUCTION

The role of a parser in computer interpretation of English is to determine the syntactic structure of English phrases and clauses. At the Naval Research Laboratory, we are developing a deterministic parser, based on the work of Marcus (1980), that can be used in interpreting military message narrative. A major goal of this work is to restrict, in a systematic way, the range of actions a parser can take. Specifically, we wish to formulate constraints that will simultaneously enhance parsing efficiency and, following Petrick (1974), permit the "expression and explanation of linguistic generalizations".

In this paper, we propose that in most cases a parser can determine the correct structures for English without looking into subconstituents except at the left corner. By "left corner", we mean the leftmost element of the constituent along with its lexical category (N, V, Adj, etc.). For example, the left corner of *They failed to inform us* is [they, pronoun]. This **Left Corner Constraint** thus restricts the parser from examining any information about a constituent other than its syntactic category (e.g. S, NP) and its left corner.

We have built this constraint into a parser that is based on the model described in Marcus (1980). The parser has two data structures: a stack of incomplete nodes and a buffer containing complete nodes that may be terminal elements or completed phrases. The buffer can contain up to three constituents. Depending on what is in the buffer and what is on top of the incomplete nodestack, the parser's pattern-action rules will start building a new constituent, declare a constituent complete, or attach a constituent to the current incomplete node. The parser is deterministic in that all structures it builds are indelible.

While it is restricted from looking at anything in a tree except the left corner, our parser still covers a wide range of English syntax. It happens that this restriction leads to a description of complementation in English that is consistent with the findings of recent linguistic theory (Fiengo 1974, Chomsky 1981). In what follows we concentrate on the issues raised by verb complementation, in particular, the problem of recognizing complement clauses. First we outline a general solution and then we describe our implementation of this solution as part of a deterministic parser of English. We include a brief discussion of adjective, noun, and preposition complementation and review three areas that seem to be exempt from the constraint. Details of the implementation are discussed in Fitzpatrick (1983) and Hindle (1983).

II VERB COMPLEMENTATION

The constituents that can occur as verb complements include the major phrase categories: noun phrases, prepositional phrases, and clauses. Because these constituents can serve in roles other than verb complement, the parser should build them in a general way without referring to verb complementation. But this means that once a phrase is built, the parser is faced with the problem of determining the phrase's syntactic relationship to other constituents, i.e. whether the phrase should be attached to the tree as a verb complement or as something else.

When the constituent is a noun phrase, determining that it is a verb complement is relatively easy since the only information that is needed about this constituent is its syntactic category, *NP*. Structural differences among NPs are not relevant to the syntactic restrictions on verb complementation. Thus if a verb is lexically marked to take a complement NP, then any unattached post-verbal NP can serve as its complement.*

*In sentences where a NP is displaced, the post-verbal complement position is filled by a NP *trace* (Fiengo, 1974). Thus *t*

Prepositional phrases are more complicated since the status of PP as a complement often depends on the preposition. For example, the PP is a complement in *agree on a plan* and *agree with everyone else*, but it is an adjunct in *agree after several hours of discussion* because *agree* admits a PP complement only if the PP begins with a particular preposition, and *after* is not on its list. In such cases, determining complement status requires the parser to discriminate among particular *types* of a syntactic category. The parser does this by examining the left terminal of PP, namely, the particular preposition.

A similar solution applies to the identification of declaratives like the *that* clause in (1a-b). The clause in (1a) is a complement to *think;* that in (1b) is an adjunct.

(1) a. I thought that I might be free of it.

b. I left that I might be free of it.

These examples are straightforward because *that* uniquely identifies a clause as a declarative and verbs can be lexically marked if they take this complement type; thus *think* is marked for a *that*-clause, but *leave* is not. In such cases, the parser decides whether or not the clause is a complement by checking the syntactic category (Sentence), the lower left terminal of the sentence *(that)*, and the lexical entry of the verb.

For infinitive phrases, however, determining complement status is more difficult. The infinitive phrase in (2a) is a complement to *fail*, that in (2b) is an adjunct. The one in (2c) is a complement to *found*.

(2) a. The drum ejector fails to function properly.

b. The drum ejector must cycle to function properly.

c. We found the drum ejector to be faulty.

Distinguishing between complements and adjuncts depends on first deciding what the internal structure of a string is, i.e. whether or not a string has the structure of a clause and, if so, then what type of clause. Infinitive phrases raise problems because they give few explicit clues to their internal structure. In (2a-b), for example, the infinitive has no overt subject and therefore bears little resemblance to clauses like that in (1a-b). Even if there is a noun that can be the subject, as in (2c), the only surface clue that identifies the infinitive is the embedded auxiliary *to*.

III CLAUSAL VERB COMPLEMENTS

Recent work in transformational theory, specifically the government and binding (GB) framework of Chomsky (1981), suggests an analysis of clausal complements that is useful for making parsing decisions. In particular, we make use of the claim that infinitives and *-ing* phrases have subjects that may be overt, as in (3a-b), or understood, as in (4a-b):

(3) a. They wanted [the contractor to make repairs]

b. This will facilitate [their making repairs]

(4) a. They attempted [to make repairs]

b. This will facilitate [making repairs]

A subject is overt or understood depending on whether it is a word listed in the lexicon or an abstract NP that is inserted into subject position by syntactic rules. In general, the abstract NP DELTA occurs as a non-lexical terminal in the context *__to VP*. The abstract possessive NP DELTA'S is a non-lexical terminal that is inserted in the context *__ing VP* (where previous syntactic rules have applied to *verb+ing* and moved the *-ing* suffix to a position preceding the verb phrase).

Abstract subjects help semantic interpretation determine the argument structure of propositions without adding new structure to the syntactic tree. For example, to identify the agent of *fix* in *He promised them to fix it*, semantic interpretation need only mark as coreferential the DELTA subject of *fix* and the matrix subject *He*, using structural constraints on the coindexing of NPs.

Because they have subjects, infinitive and *-ing* phrases are assigned the structure of a clause, analogous to the embedded declarative of (1a-b). Consequently, we now have all the properties that are needed to identify a clause type by its syntactic category and leftmost terminal. The general rule is:

> If the clause begins with a complementizer (i.e. *that*, *for*, or a *wh* phrase such as *who*, *what*, *where*, *how*), then this complementizer is the left terminal that identifies the clause type; if the clause has no complementizer, then the subject NP (which is lexical or abstract) is the lower left terminal that identifies the clause type.

We refer to clauses that contain a complementizer as S-bar nodes and assign them the structure in (5) (the COMP node contains the complementizer):

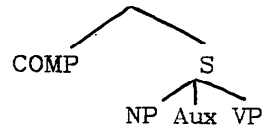
(=trace) is a 'place holder' for the interrogative NP *which* and the definite NP *the circuits* in (i) and (ii), respectively:

(i) Which circuits should we check *t*?

(ii) The circuits were checked *t* prior to installation.

Note that the parser must be prevented from attaching constituents as complements when a verb's complement positions are already filled. For example, *repair* takes a single NP as complement, as in *repair forward kingpost* and *repair new ships*. In *repair new ships forward kingpost*, therefore, *forward kingpost* cannot be interpreted as a second NP object. The problem of complement numbering involves the argument structure of verbs. Although the parser currently uses syntactic rules to keep track of the number of complements attached, we expect that this should actually be handled by semantic interpretation rules that interact with the syntax to monitor argument structure and reject attachments to filled argument positions.

(5)



Clauses without a complementizer are simply called S nodes and have the structure NP + Aux + VP.

As we observed earlier, identifying declaratives like that in (1a-b) is fairly easy because the leftmost terminal is the complementizer *that*, which uniquely specifies the clause, and because verbs can be lexically marked for taking *that* clauses as complements. *For-to* infinitives like the one in (6) are another clause type that can be identified fairly easily.

(6) We will arrange [for the shipyard to complete repairs]

In this case, the left corner is the complementizer *for*, which uniquely specifies the clause as an infinitive with a lexical subject. The verbs *arrange*, *intend*, *prefer*, and *hate*, among others, are lexically marked for taking a *for-to* complement.

Identifying *wh* clauses like those in (7a-b) is equally straightforward.

(7) a. Our investigation will indicate [whether we can repair the ring]

b. Our investigation will indicate [whether to repair the ring]

Since the *wh* words always mark the beginning of a *wh* clause and since the interior of the clause is irrelevant to its complement status-- *wh* clauses can be either declarative or infinitival--the parser only needs to examine the left corner of the clause in order to identify it correctly. This is true even if the *wh* word is embedded in a prepositional phrase, as in,

(8) The investigation will indicate [for which unit repairs should be implemented]

The only time prepositions occur in a COMP node is when they are part of a *wh* phrase like *for which, with whom, by how many days*, etc. Consequently, a clause whose dominating node is S-bar and whose leftmost terminal is lexically specified as [preposition] can always be identified correctly as a *wh* clause. The embedded clause in (8), where *for* is a preposition, is therefore distinguished from the embedded clause in (6), where *for* is lexically specified as a complementizer. It is also distinguished from the subordinate clause in *We asked for we wanted to be sure*, where *for* is a preposition, as in (8), but the dominating node is a PP with the structure P + S.

If the embedded clause has no complementizer, it has no COMP node. Therefore, the left edge of the clause will be an abstract subject, as in (8a-c), or a lexical subject, as in (8d):

(8) a. Ship's force attempted [DELTA to make repairs]

b. He promised them [DELTA to fix it]

c. They tried [DELTA'S installing a new antenna]

d. We found [the transistors were bad]

Because DELTA is only inserted in the context *___to VP*, a clause with DELTA as its lower left terminal will always be identified correctly as an infinitive. Similarly, DELTA'S is only inserted in the context *___ing VP*, so that a clause with DELTA'S in the left corner can always be identified as an -ing clause. When a clause begins with a lexical subject, however, it can be either an infinitive or a declarative; thus *it* is the subject of a declarative in (10a) but the subject of an infinitive in (10b).

(10) a. We assumed [it was inoperable].

b. We assumed [it to be inoperable].

This would be a serious problem if the distinction between declaratives and infinitives were relevant in determining complement status for verbs like *assume*. But as it happens, verbs like *assume* do not discriminate between infinitives and declaratives; the interior of the clause is irrelevant since only the subject type counts. This generalization holds for each of the verbs in (11):*

(11) assume, believe, claim, conclude, confirm, consider, demonstrate, discover, establish, feel, find, know, learn, observe, note, notice, report, say, show, suppose, think

The presence of a lexical subject in complements to the verbs in (11) thus parallels exactly the situation we described with the *wh* clauses: once the parser finds the left corner it needs no further information about the clause because the verbs that choose this complement type do not discriminate between declaratives and infinitives.

IV OTHER COMPLEMENT SYSTEMS

Data from other phrase types suggest that our account of complementation should not be limited to the verb system. Complements to adjectives, nouns, and prepositions follow patterns that parallel the ones we have just described. Each of these categories takes PP complements, e.g. *sorry for them* (Adj+PP), *a promise to them* (N+PP), and *from behind some parked cars* (P+PP). Each also takes clausal complements, although the range of clause types differs for each category. Infinitives

*Notice that the verbs *conclude*, *learn*, and *say* take an infinitive complement only in their passive form, e.g. *This is said to be a fact*, *This was learned to be a fact*, where the subject of the infinitive is a trace. Our parser assumes, following current transformational theory, that the syntax treats a trace in embedded subject position in the same way it treats a lexical subject (Chomsky 1981).

The generalization includes verbs like *seem* and *appear* if, following Marcus (1980) and recent linguistic theory, infinitive complements to these verbs are analyzed with a trace subject, like the complements to *learn* and *say*. Thus (i) and (ii) are analogous to (10a) and (10b), respectively:

(i) It seems [the transistors are bad]

(ii) The transistors seem [t to be bad]

with a lexical subject and no complementizer only occur with verbs. For example, the verb *assume* takes an infinitive complement in *We assumed it to be inoperable* but the noun *assumption* does not; *our assumption it to be inoperable* is not a possible N + S expression (although *that* clauses occur with both V and N, e.g. *We assumed that it was inoperable, our assumption that it was inoperable).* Nouns and adjectives do, however, take *for-to* and DELTA-subject complements; *eager for them to leave* and *eager to leave* are APs, *the plans for them to meet* and *the plans to meet* are NPs. Our investigations, though not yet complete, thus support a description of complementation in NPs, APs, and PPs that is consistent with the claims of the Left Corner Constraint.

V IMPLICATIONS OF THE CONSTRAINT

It has often been assumed that, in order to make the correct decisions about a constituent, a parser must have access to certain information about the constituent's internal structure. Methods of providing this information explicitly include the "annotated surface structures" of Winograd (1972) and Marcus (1980), where nodes contain bundles of features that specify certain properties of internal structure, e.g. whether a clause is declarative or infinitive. The Left Corner Constraint introduces a new approach by claiming that all relevant information about internal structure can be inferred from the leftmost terminal of a constituent. The use of additional devices to record syntactic structure thus becomes unnecessary when the parser incorporates this constraint together with appropriate grammatical formalisms.

In some cases, however, features and other explicit devices are needed. We know of three: the attachment to COMP of a PP containing a *wh* phrase, the agreement between heads of a phrase, and the recognition of idioms. Specifically, when PP contains a *wh* feature, a COMP node can be created only after the *wh* feature has been percolated up to the PP node from an internal position that can be deeply embedded, e.g. *in how many days, from behind which cars.* Agreement requires that features like "plural", "singular", and "human" be projected onto a phrase node (e.g. a subject NP) and matched against features on other phrase nodes. Idioms like *make headway* in *we made substantial headway* also require the parser to have access to more than the left corner.

Each of these cases involves special complications that may explain why they are exempt from the Left Corner Constraint. In a PP, these complications have to do with the depth of embedding of a *wh* word and with the optionality of preposition stranding (as in *They need to know which units to look into).* Complications from semantic interpretation arise with agreement patterns, which depend on selection as well as syntactic features, and with idioms, which reflect the interaction of syntactic structure and metaphor.

VI CONCLUSIONS

Our discussion of the Left Corner Constraint has focussed on results obtained in our studies of verb complementation, in particular, clausal complements. We have shown that, given certain concepts from the GB framework, the constraint enhances parsing efficiency because it allows the parser to infer properties of internal structure from the leftmost terminal; the parser can therefore avoid mentioning these properties explicitly. Specifically, we have shown that:

(1) Within a complement category (e.g. prepositional phrase, clause), complement types can be distinguished according to their leftmost terminal.

(2) The leftmost terminal of a clausal complement is always a complementizer or a subject NP, even if the complement is a 'subjectless' clause. Clause types are therefore distinguished by their complementizer or by their subject NP, which is either a lexical item or one of the abstract NPs DELTA or DELTA'S.

(3) Verbs discriminate among clause types according to the leftmost terminal of a clause. Hence, the distinctions between tensed and infinitival clauses is important in verb complementation only when it coincides with the distinctions among left corner elements.

(4) The Left Corner Constraint can lead to a more general description of complementation that includes the complement system of adjectives, nouns, and prepositions.

ACKNOWLEDGEMENTS

We would like to thank Ralph Grishman, Constance Heitmeyer, and Stanley Wilson for many helpful comments on this paper.

REFERENCES

Chomsky, N. 1981. *Lectures on Government and Binding.* Dordrecht: Foris Publications.

Fiengo, R. 1974. On Trace Theory. *Linguistic Inquiry.* vol. 8, no. 1.

Fitzpatrick, E. 1983. Verb Complements in a Deterministic Parser. NRL Technical Memorandum #7590-077, March 1983.

Hindle, D. 1983. User Manual for Fidditch, A Deterministic Parser. NRL Technical Memorandum #7590-142. June 1983.

Marcus, M. 1980. *A Theory of Syntactic Recognition for Natural Language.* Cambridge, MA: MIT Press.

Petrick., S. R. 1974. Review of Winograd, "A Procedural Model of Natural Language Understanding." *Computing Reviews,* 15.

Winograd, T. 1972. *Understanding Natural Language.* New York: Academic Press.