Deterministic Parsing of Syntactic Non-fluencies

Donald Hindle

Bell Laboratories
Murray Hill, New Jersey 07974

It is often remarked that natural language, used naturally, is unnaturally ungrammatical.* Spontaneous speech contains all manner of false starts, hesitations, and self-corrections that disrupt the well-formedness of strings. It is a mystery then, that despite this apparent wide deviation from grammatical norms, people have little difficulty understanding the non-fluent speech that is the essential medium of everyday life. And it is a still greater mystery that children can succeed in acquiring the grammar of a language on the basis of evidence provided by a mixed set of apparently grammatical and ungrammatical strings.

1. Self-correction: a Rule-governed System

In this paper I present a system of rules for resolving the non-fluencies of speech, implemented as part of a computational model of syntactic processing. The essential idea is that non-fluencies occur when a speaker corrects something that he or she has already said out loud. Since words once said cannot be unsaid, a speaker can only accomplish a self-correction by saying something additional -- namely the intended words. The intended words are supposed to substitute for the wrongly produced words. For example, in sentence (1), the speaker initially said *I* but meant *we*.

(1) I was-- we were hungry.

The problem for the hearer, as for any natural language understanding system, is to determine what words are to be expunged from the actual words said to find the intended sentence.

Labov (1966) provided the key to solving this problem when he noted that a phonetic signal (specifically, a markedly abrupt cut-off of the speech signal) always marks the site where self-correction takes place. Of course, finding the site of a self-correction is only half the problem; it remains to specify what should be removed. A first guess suggests that this must be a non-deterministic problem, requiring complex reasoning about what the speaker meant to say. Labov claimed that a simple set of rules operating on the surface string would specify exactly what should be changed, transforming nearly all non-fluent strings into fully grammatical sentences. The specific set of transformational rules Labov proposed were not formally adequate, in part because they were surface transformations which ignored syntactic constituenthood. But his work forms the basis of this current analysis.

* This research was done for the most part at the University of Pennsylvania, supported by the National Institute of Education under grants G78-0169 and G80-0163.

Labov's claim was not of course that ungrammatical sentences are never produced in speech, for that clearly would be false. Rather, it seems that truly ungrammatical productions represent only a tiny fraction of the spoken output, and in the preponderance of cases, an apparent ungrammaticality can be resolved by simple editing rules. In order to make sense of non-fluent speech, it is essential that the various types of grammatical deviation be distinguished.

This point has sometimes been missed, and fundamentally different kinds of deviation from standard grammaticality have been treated together because they all present the same sort of problem for a natural language understanding system. For example, Hayes and Mouradian (1981) mix together speaker-initiated self-corrections with fragmentary sentences of all sorts:

> people often leave out or repeat words or phrases, break off what they are saying and rephrase or replace it, speak in fragments, or otherwise use incorrect grammar (1981:231).

Ultimately, it will be essential to distinguish between non-fluent productions on the one hand, and constructions that are fully grammatical though not yet understood, on the other. Although we may not know in detail the correct characterization of such processes as ellipsis and conjunction, they are without doubt fully productive grammatical processes. Without an understanding of the differences in the kinds of non-fluencies that occur, we are left with a kind of grab bag of grammatical deviation that can never be analyzed except by some sort of general purpose mechanisms.

In this paper, I want to characterize the subset of spoken non-fluencies that can be treated as self-corrections, and to describe how they are handled in the context of a deterministic parser. I assume that a system for dealing with self-corrections similar to the one I describe must be a part of the competence of any natural language user. I will begin by discussing the range of non-fluencies that occur in speech. Then, after reviewing the notion of deterministic parsing, I will describe the model of parsing self-corrections in detail, and report results from a sample of 1500 sentences. Finally, I discuss some implications of this theory of self-correction, particularly for the problem of language acquisition.

2. Errors in Spontaneous Speech

Linguists have been of less help in describing the nature of spoken non-fluencies than might have been hoped; relatively little attention has been devoted to the actual performance of speakers, and studies that claim to be based

on performance data seem to ignore the problem of non-fluencies. (Notable exceptions include Fromkin (1980), and Thompson (1980)). For the discussion of self-correction, I want to distinguish three types of non-fluencies that typically occur in speech.

1. Unusual Constructions. It is perhaps worth emphasizing that the mere fact that a parser does not handle a construction, or that linguists have not discussed it, does not mean that it is ungrammatical. In speech, there is a range of more or less unusual constructions which occur productively (some occur in writing as well), and which cannot be considered syntactically ill-formed. For example,

(2a) I imagine there's a lot of them must have had some good reasons not to go there.
(2b) That's the only thing he does is fight.

Sentence (2a) is an example of non-standard subject relative clauses that are common in speech. Sentence (2b), which seems to have two tensed "be" verbs in one clause is a productive sentence type that occurs regularly, though rarely, in all sorts of spoken discourse (see Kroch and Hindle 1981). I assume that a correct and complete grammar for a parser will have to deal with all grammatical processes, marginal as well as central. I have nothing further to say about unusual constructions here.

2. True Ungrammaticalities. A small percentage of spoken utterances are truly ungrammatical. That is, they do not result from any regular grammatical process (however rare), nor are they instances of successful self-correction. Unexceptionable examples are hard to find, but the following give the flavor.

(3a) I've seen it happen is two girls fight.

(3b) Today if you beat a guy wants to blow your head off for something.

(3c) And aa a lot of the kids that are from our neighborhood-- there's one section that the kids aren't too-- think they would usually-- the-- the ones that were the-- the drop outs and the stoneheads.

Labov (1966) reported that less that 2% of the sentences in a sample of a variety of types of conversational English were ungrammatical in this sense, a result that is confirmed by current work (Kroch and Hindle 1981).

3. Self-corrected strings. This type of non-fluency is the focus of this paper. Self-corrected strings all have the characteristic that some extraneous material was apparently inserted, and that expunging some substring results in a well-formed syntactic structure, which is apparently consistent with the meaning that is intended.

In the degenerate case, self-correction inserts non-lexical material, which the syntactic processor ignores, as in (4).

(4a) He was uh still asleep.
(4b) I didn't ko-- go right into college.

The minimal non-lexical material that self-correction might insert is the editing signal itself. Other cases (examples 6-10 below) are only interpretable given the assumption that certain words, which are potentially part of the syntactic structure, are to be removed from the syntactic analysis.

The status of the material that is corrected by self-correction and is expunged by the editing rules is somewhat odd. I use the term *expunction* to mean that it is removed from any further syntactic analysis. This does not mean however that a self-corrected string is unavailable for *semantic* processing. Although the self-corrected string is edited from the syntactic analysis, it is nevertheless available for semantic interpretation. Jefferson (1974) discusses the example

(5) ... [thuh] -- [thiy] officer ...

where the initial, self-corrected string (with the pre-consonantal form of *the* rather than the pre-vocalic form) makes it clear that the speaker originally intended to refer to the police by some word other than *officer*.

I should also note that the problems addressed by the self-correction component that I am concerned with are only part of the kind of deviance that occurs in natural language use. Many types of naturally occurring errors are not part of this system, for example, phonological and semantic errors. It is reasonable to hope that much of this dreck will be handled by similar subsystems. Of course, there will always remain errors that are outside of any system. But we expect that the apparent chaos is much more regular than it at first appears and that it can be modeled by the interaction of components that are themselves simple.

In the following discussion, I use the terms *self-correction* and *editing* more or less interchangeably, though the two terms emphasize the generation and interpretation aspects of the same process.

3. The Parser

The editing system that I will describe is implemented on top of a deterministic parser, called *Fidditch*, based on the processing principles proposed by Marcus (1980). It takes as input a sentence of standard words and returns a labeled bracketing that represents the syntactic structure as an annotated tree structure. Fidditch was designed to process transcripts of spontaneous speech, and to produce an analysis, partial if necessary, for a large corpus of interview transcripts. Because it is a deterministic parser, it produces only one analysis for each sentence. When Fidditch is unable to build larger constituents out of subphrases, it moves on to the next constituent of the sentence.

In brief, the parsing process proceeds as follows. The words in a transcribed sentence (where sentence means one tensed clause together with all subordinate clauses) are assigned a lexical category (or set of lexical categories) on the basis of a 2000 word lexicon and a morphological analyzer. The lexicon contains, for each word, a list of possible lexical categories, subcategorization information, and in a few cases, information on compound words. For example, the entry for *round* states that it is a noun, verb, adjective or preposition, that as a verb it is subcategorized for the movable particles *out* and *up* and for *NP*, and that it may be part of the compound adjective/preposition *round about*.

Once the lexical analysis is complete, The phrase structure tree is constructed on the basis of pattern-action rules using two internal data structures: 1) a push-down stack of incomplete nodes, and 2) a buffer of complete constituents, into which the grammar rules can look through

a window of three constituents. The parser matches rule patterns to the configuration of the window and stack. Its basic actions include

— starting to build a new node by pushing a category onto the stack

— attaching the first element of the window to the stack

— dropping subtrees from the stack into the first position in the window when they are complete.

The parser proceeds deterministically in the sense that no aspect of the tree structure, once built may be altered by any rule. (See Marcus 1980 for a comprehensive discussion of this theory of parsing.)

4. The self-correction rules

The self-correction rules specify how much, if anything, to expunge when an editing signal is detected. The rules depend crucially on being able to recognize an editing signal, for that marks the right edge of an expunction site. For the present discussion, I will assume little about the phonetic nature of the signal except that it is phonetically recognizable, and that, whatever their phonetic nature, all editing signals are, for the self-correction system, equivalent. Specifying the nature of the editing signal is, obviously, an area where further research is needed.

The only action that the editing rules can perform is *expunction*, by which I mean removing an element from the view of the parser. The rules never replace one element with another or insert an element in the parser data structures. However, both replacements and insertions can be accomplished within the self-correction system by expunction of partially identical strings. For example, in

(6) I am-- I was really annoyed.

The self-correction rules will expunge the *I am* which precedes the editing signal, thereby in effect replacing *am* with *was* and inserting *really*.

Self-corrected strings can be viewed formally as having extra material inserted, but not involving either deletion or replacement of material. The linguistic system does seem to make use of both deletions and replacements in other subsystems of grammar however, namely in ellipsis and rank shift. As with the editing system, these are not errors but formal systems that interact with the central features of the syntax. True errors do of course occur involving all three logical possibilities (insertion, deletion, and replacement) but these are relatively rare.

The self-correction rules have access to the internal data structures of the parser, and like the parser itself, they operate deterministically. The parser views the editing signal as occurring at the *end* of a constituent, because it marks the *right* edge of an expunged element. There are two types of editing rules in the system: expunction of copies, for which there are three rules, and lexically triggered restarts, for which there is one rule.

4.1 Copy Editing

The copying rules say that if you have two elements which are the same and they are separated by an editing signal, the first should be expunged from the structure. Obviously the trick here is to determine what counts as copies. There are three specific places where copy editing applies.

SURFACE COPY EDITOR. This is essentially a non-syntactic rule that matches the surface string on either side of the editing signal, and expunges the first copy. It applies to the surface string (i.e., for transcripts, the orthographic string) before any syntactic processing. For example, in (7), the underlined strings are expunged before parsing begins.

(7a) Well *if they'd*-- if they'd had a knife *I wou*-- I wouldn't be here today.
(7b) *If they*-- if they could do it.

Typically, the Surface Copy Editor expunges a string of words that would later be analyzed as a constituent (or partial constituent), and would be expunged by the Category or the Stack Editors (as in 7a). However, the string that is expunged by the Surface Copy Editor need not be dominated by a single node; it can be a sequence of unrelated constituents. For example, in (7b) the parser will not analyze the first *if they* as an SBAR node since there is no AUX node to trigger the start of a sentence, and therefore, the words will not be expunged by either the Category or the Stack editor. Such cases where the Surface Copy Editor *must* apply are rare, and it may therefore be that there exists an optimal parser grammar that would make the Surface Copy Editor redundant; all strings would be edited by the syntactically based Category and Stack Copy rules. However, it seems that the Surface Copy Editor must exist at some stage in the process of syntactic acquisition. The overlap between it and the other rules may be essential in learning.

CATEGORY COPY EDITOR. This copy editor matches syntactic constituents in the first two positions in the parser's buffer of complete constituents. When the first window position ends with an editing signal and the first and second constituents in the window are of the same type, the first is expunged. For example, in sentence (8) the first of two determiners separated by an editing signal is expunged and the first of two verbs is similarly expunged.

(8) I was just *that* -- the kind of guy that didn't *have*-- like to have people worrying.

STACK COPY EDITOR. If the first constituent in the window is preceded by an editing signal, the Stack Copy Editor looks into the stack for a constituent of the same type, and expunges any copy it finds there along with all descendants. (In the current implementation, the Stack Copy Editor is allowed to look at successive nodes in the stack, back to the first COMP node or attention shifting boundary. If it finds a copy, it expunges that copy along with any nodes that are at a shallower level in the stack. If Fidditch were allowed to attach of incomplete constituents, the Stack Copy Editor could be implemented to delete the copy only, without searching through the stack. The specifics of the implementation seems not to matter for this discussion of the editing rules.) In sentence (9), the initial embedded sentence is expunged by the Stack Copy Editor.

(9) I think that *you get*-- it's more strict in Catholic schools.

4.2 An Example

It will be useful to look a little more closely at the operation of the parser to see the editing rules at work. Sentence (10)

(10) I-- the-- the guys that I'm-- was telling you about were.

includes three editing signals which trigger the copy editors. (note also that the complement of *were* is ellipted.) I will show a trace of the parser at each of these correction stages.

The first editor that comes into play is the Surface Copy Editor, which searches for identical strings on either side of an editing signal, and expunges the first copy. This is done once for each sentence, before any lexical category assignments are made. Thus in effect, the Surface Copy Editor corresponds to a phonetic/phonological matching operation, although it is in fact an orthographic procedure because we are dealing with transcriptions. Obviously, a full understanding of the self-correction system calls for detailed phonetic/phonological investigations.

After the Surface Copy Editor has applied, the string that the lexical analyzer sees is (11)

(11) I-- the guys that I'm-- was telling you about were.

rather than (10). Lexical assignments are made, and the parser proceeds to build the tree structures. After some processing, the configuration of the data structures is that shown in Figure 1.

NODE STACK

5	NP < I -- >
4	NP < the guys >
3	>>ATTENSHIFT<<
2	NP < I >
current	AUX < am -- >

COMPLETE NODES IN WINDOW

AUX < was >	V < telling >	PRON < you >

Figure 1. The parser state before the Stack Copy Editor applies.

NODE STACK

4	NP < I -- >
3	NP < the guys >
2	>>ATTENSHIFT<<
current	NP < I >

COMPLETE NODES IN WINDOW

AUX < was >	V < telling >	PRON < you >

Figure 2. The parser state after Stack Copy Editing the AUX node.

Before determining what next rule to apply, the two editing rules come into play, the Category Editor and the Stack Editor. At this pulse, the Stack Editor will apply because the first constituent in the window is the same (an AUX node) as the current active node, and the current node ends with an edit signal. As a result, the first window element is popped into another dimension, leaving the the parser data structures in the state shown in Figure 2.

Parsing of the sentence proceeds, and eventually reaches the state shown in Figure 3. where the Stack Editor conditions are again met. The current active node and the first element in the window are both NPs, and the active node ends with an edit signal. This causes the current node to be expunged, leaving only a single NP node, the one in the window. The final analysis of the sentence, after some more processing is the tree shown in Figure 4.

I should reemphasize that the status of the edited elements is special. The copy editing rules remove a constituent, no matter how large, from the view of the parser. The parser continues as if those words had not been said. Although the expunged constituents may be available for semantic interpretation, they do not form part of the main predication.

NODE STACK

current	NP < I -- >

COMPLETE NODES IN WINDOW

NP < the guys >	SBAR < that... >	AUX < were >

Figure 3. The parser state before the second application of the Stack Copy Editor.

```
S
  NP NP
       DETER DART the
       NOM N
            pl
            N guy
    SBAR
       COMP
          CMP that
          NP t
       S
         NP PRON I
         AUX
            TNS PAST s
            be
            +ing
         VP
            V tell
            NP PRON you
            PREP about
            NP t
AUX TNS PAST pl
VP V be
```

Figure 4. The final analysis of sentence (10).

4.3 Restarts

A somewhat different sort of self-correction, less sensitive to syntactic structure and flagged not only by the editing signal but also by a lexical item, is the *restart*. A restart triggers the expunction of all words from the edit signal back to the beginning of the sentence. It is signaled by a standard edit signal followed by a specific lexical item drawn from a set including *well, ok, see, you know, like I said*, etc. For example,

(12a) *That's the way if--* well everybody was so stoned, anyway.
(12b) *But when I was young I went in--* oh I was nineteen years old.

It seems likely that, in addition to the lexical signals, specific intonational signals may also be involved in restarts.

5. A sample

The editing system I have described has been applied to a corpus of over twenty hours of transcribed speech, in the process of using the parser to search for various syntactic constructions. The transcripts are of sociolinguistic interviews of the sort developed by Labov and designed to elicit unreflecting speech that approximates natural conversation.* They are conversational interviews covering a range of topics, and they typically include considerable non-fluency. (Over half the sentences in one 90 minute interview contained at least one non-fluency).

The transcriptions are in standard orthography, with sentence boundaries indicated. The alternation of speakers' turns is indicated, but overlap is not. Editing signals, when noted by the transcriber, are indicated in the transcripts with a double dash. It is clear that this approach to transcription only imperfectly reflects the phonetics of editing signals; we can't be sure to what extent the editing signals in our transcripts represent facts about production and to what extent they represent facts about perception. Nevertheless, except for a general tendency toward underrepresentation, there seems to be no systematic bias in our transcriptions of the editing signals, and therefore our findings are not likely to be undone by a better understanding of the phonetics of self-correction.

One major problem in analyzing the syntax of English is the multiple category membership of words. In general, most decisions about category membership can be made on the basis of local context. However, by its nature, self-correction disrupts the local context, and therefore the disambiguation of lexical categories becomes a more difficult problem. It is not clear whether the rules for category disambiguation extend across an editing signal or not. The results I present depend on a successful disambiguation of the syntactic categories, though the algorithm to accomplish this is not completely specified. Thus, to test the self-correction routines I have, where necessary, imposed the proper category assignment.

Table 1 shows the result of this editing system in the parsing of the interview transcripts from one speaker. All in all this shows the editing system to be quite successful in resolving non-fluencies.

* The interviews for this study were conducted by Tony Kroch and by Anne Bower.

TABLE 1. SELF-CORRECTION RULE APPLICATION

total sentences	1512
total sentences with no edit signal	1108 (73%)

Editing Rule	Applications	
expunction of edit signal only	128	24%
surface copy	161	29%
category copy	47	9%
stack copy	148	27%
restart	32	6%
failures	17	3%
remaining unclear and ungrammatical	11	2%

6. Discussion

Although the editing rules for Fidditch are written as deterministic pattern-action rules of the same sort as the rules in the parsing grammar, their operation is in a sense isolable. The patterns of the self-correction rules are checked first, before any of the grammar rule patterns are checked, at each step in the parse. Despite this independence in terms of rule ordering, the operation of the self-correction component is closely tied to the grammar of the parser; for it is the parsing grammar that specifies what sort of constituents count as the same for copying. For example, if the grammar did not treat *there* as a noun phrase when it is subject of a sentence, the self-correction rules could not properly resolve a sentence like

(13) *People--* there's a lot of people from Kennsington

because the editing rules would never recognize that *people* and *there* are the same sort of element. (Note that (13) cannot be treated as a Restart because the lexical trigger is not present.) Thus, the observed pattern of self-correction introduces empirical constraints on the set of features that are available for syntactic rules.

The self-correction rules impose constraints not only on what linguistic elements must count as the same, but also on what must count as different. For example, in sentence (14), *could* and *be* must be recognized as different sorts of elements in the grammar for the AUX node to be correctly resolved. If the grammar assigned the two words exactly the same part of speech, then the Category Copy Editor would necessarily apply, incorrectly expunging *could*.

(14) Kid could-- be a brain in school.

It appears therefore that the pattern of self-corrections that occur represents a potentially rich source of evidence about the nature of syntactic categories.

Learnability. If the patterns of self-correction count as evidence about the nature of syntactic categories for the linguist, then this data must be equally available to the language learner. This would suggest that, far from being an impediment to language learning, non-fluencies may in fact facilitate language acquisition by highlighting equivalent classes.

This raises the general question of how children can acquire a language in the face of unrestrained non-fluency. How can a language learner sort out the grammatical from the ungrammatical strings? (The non-fluencies of speech are of course but one aspect of the degeneracy of input that makes language acquisition a puzzle.) The self-correction system I have described suggests that many non-fluent strings can be resolved with little detailed linguistic knowledge.

As Table 1 shows, about a quarter of the editing signals result in expunction of only non-linguistic material. This requires only an ability to distinguish linguistic from non-linguistic stuff, and it introduces the idea that edit signals signal an expunction site. Almost a third are resolved by the Surface Copying rule, which can be viewed simply as an instance of the general non-linguistic rule that multiple instances of the same thing count as a single instance. The category copying rules are generalizations of simple copying, applied to a knowledge of linguistic categories. Making the transition from surface copies to category copies is aided by the fact that there is considerable overlap in coverage, defining a path of expanding generalization. Thus at the earliest stages of learning, only the simplest, non-linguistic self-correction rules would come into play, and gradually the more syntactically integrated would be acquired.

Contrast this self-correction system to an approach that handles non-fluencies by some general problem solving routines, for example Granger (1982), who proposes reasoning from what a speaker might be expected to say. Besides the obvious inefficiencies of general problem solving approaches, it is worth giving special emphasis to the problem with learnability. A general problem solving approach depends crucially on evaluating the likelihood of possible deviations from the norms. But a language learner has by definition only partial and possibly incorrect knowledge of the syntax, and is therefore unable to consistently identify deviations from the grammatical system. With the editing system I describe, the learner need not have the ability to recognize deviations from grammatical norms, but merely the non-linguistic ability to recognize copies of the same thing.

Generation. Thus far, I have considered the self-correction component from the standpoint of parsing. However, it is clear that the origins are in the process of generation. The mechanism for editing self-corrections that I have proposed has as its essential operation expunging one of two identical *elements*. It is unable to expunge a sequence of two elements. (The Surface Copy Editor might be viewed as a counterexample to this claim, but see below.) Consider expunction now from the standpoint of the generator. Suppose self-correction bears a one-to-one relationship to a possible action of the generator (initiated by some monitoring component) which could be called ABANDON CONSTRUCT X. And suppose that this action can be initiated at any time up until CONSTRUCT X is completed, when a signal is returned that the construction is complete. Further suppose that ABANDON CONSTRUCT X causes an editing signal. When the speaker decides in the middle of some linguistic element to abandon it and start again, an editing signal is produced.

If this is an appropriate model, then the elements which are self-corrected should be exactly those elements that exist at some stage in the generation process. Thus, we should be able to find evidence for the units involved in generation by looking at the data of self-correction. And indeed, such evidence should be available to the language learner as well.

Summary

I have described the nature of self-corrected speech (which is a major source of spoken non-fluencies) and how it can be resolved by simple editing rules within the context of a deterministic parser. Two features are essential to the self-correction system: 1) every self-correction site (whether it results in the expunction of words or not) is marked by a phonetically identifiable signal placed at the right edge of the potential expunction site; and 2) the expunged part is the left-hand member of a pair of copies, one on each side of the editing signal. The copies may be of three types: 1) identical surface strings, which are edited by a matching rule that applies before syntactic analysis begins; 2) complete constituents, when two constituents of the same type appear in the parser's buffer; or 3) incomplete constituents, when the parser finds itself trying to complete a constituent of the same type as a constituent it has just completed. Whenever two such copies appear in such a configuration, and the first one ends with an editing signal, the first is expunged from further analysis. This editing system has been implemented as part of a deterministic parser, and tested on a wide range of sentences from transcribed speech. Further study of the self-correction system promises to provide insights into the units of production and the nature of linguistic categories.

Acknowledgements

My thanks to Tony Kroch, Mitch Marcus, and Ken Church for helpful comments on this work.

References

Fromkin, Victoria A. ed. 1980. *Errors in Linguistic Performance: Slips of the Tongue, Ear, Pen and Hand.* Academic Press: New York.

Granger, Richard H. 1982. Scruffy Text Understanding: Design and Implementation of 'Tolerant' Understanders. *Proceedings of the 20th Annual Meeting of the ACL.*

Hayes, Philip J. and George V. Mouradian. 1981. Flexible Parsing. *American Journal of Computational Linguistics* 7.4, 232-242.

Jefferson, Gail. 1974. Error correction as an interactional resource. *Language in Society* 2:181-199.

Kroch, Anthony and Donald Hindle. 1981. *A quantitative study of the syntax of speech and writing.* Final report to the National Institute of Education, grant 78-0169.

Labov, William. 1966. On the grammaticality of everyday speech. Paper presented at the Linguistic Society of America annual meeting.

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

Thompson, Bozena H. 1980. A linguistic analysis of natural language communication with computers. *Proceedings of the eighth international conference on computational linguistics.*