Resolving Lexical Ambiguity in a Deterministic Parser

Robert Milne

Intelligent Applications
10 Charlotte Square
Edinburgh EH2 4DR Scotland

Lexical ambiguity and especially part-of-speech ambiguity is the source of much non-determinism in parsing. As a result, the resolution of lexical ambiguity presents deterministic parsing with a major test. If deterministic parsing is to be viable, it must be shown that lexical ambiguity can be resolved easily deterministically. In this paper, it is shown that Marcus's "diagnostics" can be handled without any mechanisms beyond what is required to parse grammatical sentences and reject ungrammatical sentences. It is also shown that many other classes of ambiguity can be easily resolved as well.

1 Introduction

Lexical ambiguity, and especially part-of-speech ambiguity, is the source of much non-determinism in parsing. As a result, the resolution of lexical ambiguity presents deterministic parsing (Marcus 1980) with a major test. If deterministic parsing is to be viable, it should be shown that lexical ambiguity can be resolved deterministically for many situations in which people do not have trouble. In this paper, it is shown that Marcus's "diagnostics" can be handled without any mechanisms beyond what is required to parse grammatical sentences and reject ungrammatical sentences and that many other classes of ambiguity can be easily resolved as well. This result is possible because of the constraints on English from word order and number agreement.

Although many high-level constituents can be "moved" in English, the lower-level structure of some constituents is relatively fixed. For example, after a determiner, one expects a noun rather than a verb. In this paper we also wish to ask, "How might this low-level fixed order assist in the resolution of ambiguity?" We will not give a definite answer to this question, but will see that it is extremely useful in the resolution of ambiguity.

The examples of ambiguity shown in this paper seem to cause no apparent problems to a person reading them. That is, all of these examples read easily and certainly do not exhibit the garden path effect, except, of course, the examples that are intended to be difficult. If a parser is to be psychologically plausible, then it is desirable that it handle these examples in such a way as to explain why people have no apparent difficulty with most sentences, despite the inherent ambiguity in them.

In parsing English, one of the major causes of non-determinism is part-of-speech ambiguity. If a word can be two parts of speech, then a non-deterministic parser may have to explore both possibilities. If one claims to be able to parse English deterministically, then the resolution of part-of-speech ambiguity is a very important area.

It should be noted that a non-deterministic parser does not need to tackle the problem of local part-of-speech ambiguity. If it should make an error, then it can backtrack and correct it. Alternatively, it could maintain all possible parses at once and throw some of them away. In deterministic parsing we are not allowed to use either backtracking or parallelism. Although this problem has been investigated for many non-deterministic parsers, it has not been the critical problem that it is for deterministic parsing. To handle ambiguity deterministically, we must never make an error. As a result, our methods of disambiguation must be reliable. We will see that many cases of ambiguity can be resolved using standard techniques that have been applied to non-deterministic parsers.

If it is possible to handle all the examples of local ambiguity presented here, with no additional mechanism, device or feature than is needed for ordinary sentence parsing, then our goal above can be considered met. One

0362-613X/86/010001-12$03.00

possible explanation for the fact that people do not notice local ambiguities may be that there is no special mechanism needed for them, so that nothing differing from normal parsing is necessary.

Conversely, if it is necessary to add special mechanisms and routines to the parser just to handle these examples of ambiguity, then this will not explain how people can understand these examples so well and it can be considered a weakness in the model.

To say part-of-speech ambiguity can be handled deterministically but with the use of special mechanisms would be no surprise and not very important. To say one can handle part-of-speech ambiguity deterministically with no special mechanisms is a more significant claim. In this paper it is indeed suggested that many cases of part-of-speech ambiguity can be handled by the parser with no special mechanisms.

This paper is a summary of a section of the author's Ph.D. thesis (Milne 1983) with the same title and describes work done at the University of Edinburgh. That thesis presents ROBIE, a deterministic parser that is able to resolve lexical ambiguities and that is fully implemented in PROLOG. ROBIE has two lookahead buffers and does not use Marcus's Attention Shift mechanism. This means that ROBIE scans the current token and one more of lookahead. PARSIFAL scanned the current token and two lookahead cells. In this paper, only local ambiguities are addressed, that is, ambiguities that can be resolved within the sentence. Global ambiguities, which require context to resolve, are not discussed. For this paper, it is assumed that the reader is familiar with deterministic parsing and no other understanding of specific parsing mechanisms is assumed.

In the rest of this paper, we look at lexical ambiguity from simple examples to more complex ones. We start with how words are defined within the parser to be ambiguous and how the morphology can be used to resolve ambiguities. Next we look at how word order and finally various types of agreement can be used to resolve most remaining ambiguities.

2 Syntactic Context

2.1 WORD DATA STRUCTURES

As a first approach to handling ambiguity, it was asked, "If we construct a compound lexical entry for each word composed of the features of each part of speech the word can have and make no alterations to the grammar, how wide a coverage of examples will we get?"

This approach was used by Winograd (1972) and was found to be very effective for the following reason. Each word has all the possible relevant features for it. Therefore, the test will succeed for each possible part of speech with which a word can be used. In this way, all applicable rules will match. It may be that often only one rule will match, or that the first rule tried is the correct rule. The question is, how often will the rule that matches be the correct rule?

All words in ROBIE are defined in the syntactic dictionaries. Each word has a compound lexical entry incorporating all the features for all the possible parts of speech the word could have. This is exactly as was done by Winograd (1972). For example, *block* is defined as a noun and a verb, *can* is defined as a noun, auxiliary verb, and verb, and *hit* is defined as a noun and a verb. The features for each of these parts of speech are kept in the dictionary and, when the word is looked up, they are returned as a single ordered list of features. These features are sub-grouped according to the part of speech they are associated with. Hence, when the word *block* is looked up, the result returned is both the noun and the verb definition. In this way, all possibilities are returned.

In the English language, most words can have several parts of speech. This fact must be reflected in a parser of English and we do this with the multiple meanings above. When the parser has enough information to decide which is the correct part of speech, it ignores (removes) the other possibilities. In this way, we have not built structure that is later thrown away. Although some may argue that this is a form of parallelism, it seems necessary since it reflects the inherent parallelism of language.

2.2 MORPHOLOGY

The first part of the disambiguation process takes place in the morphology. When ROBIE identifies a word that has a morphological ending, the morphology must adjust the features of the word. For example, when *blocked* is identified, the feature "ed" must be added to the list of features for *block*. At the same time, a portion of the disambiguation takes place. If *block* is defined as both a noun and a verb, then *blocked* is not a noun. The morphology causes some features to be added, such as "ed, past" and some features to be removed such as "tenseless." As features that are no longer applicable are removed, so also are parts of speech and their associated features that are no longer applicable. For *blocked*, the features "noun, ns, n3p" will be removed and the features "adjective, ed, past" will be added.

The morphology will identify words such as adverbs, adjectives, and verbs in a similar way. The morphology used is very similar to that of Winograd (1972) and of Dewar, Bratley, and Thorne (1969); the part-of-speech additions and deletions are taken from Marcus (1980). Although this technique may seem obvious, it is included to point out that a majority of the occurrences of part-of-speech ambiguity can be resolved or reduced on the basis of the morphology alone.

2.3 DISAMBIGUATION

Now that we have allowed words to have multiple parts of speech and the morphology can be used to trim some of the ambiguity, we need a simple technique for disambiguating words to a single part of speech. Again, referring to Occam's Razor, what is preferable is a simple and general technique for all types of disambiguation.

In ROBIE each rule matches the features of one or two buffer cells. (The word buffer will be used interchangeably with cell. That is, **buffer** and **cell** are the same concept.) If the word *block* is in the first buffer cell, then a pattern [noun] or a pattern [verb] will match. These patterns do not relate to the other possible definitions of a word. If a rule pattern has matched on the feature "noun" in the first buffer cell, then ROBIE assumes that this word is a noun. It would then be appropriate to disambiguate the word as a noun. This is exactly as in Winograd (1972).

In a non-deterministic parser, it is not essential to find the correct rule first. If the parser runs an incorrect rule, the parser may backtrack and change the category assignment. But in a deterministic parser, there will never be any backtracking, and this solution cannot be used.

Since ROBIE does not backtrack, disambiguating the word when the pattern matches will always result in the same disambiguation as if the word were disambiguated in the grammar rule. Once a rule runs assuming a buffer contains a certain part of speech, it must be used as such in the parser. The general disambiguation scheme is: if a full pattern matches a word as a certain part of speech, then it is disambiguated as that part of speech.

The compound lexical entries and pattern-matching disambiguation alone will handle many examples of ambiguity. In the rest of this paper we see just what this can do for us.

2.4 AN EXAMPLE

Given the above mechanisms – multiple definition and disambiguation by the pattern matching, let us see how a few simple examples are handled. Consider:

(1) The falling block needs painting.

We will look only at the words *falling* and *block* in this example. The word *falling* is defined as a verb and an adjective in the dictionary and *block* is defined as a noun and a verb.

While parsing this example, after the word *the* has initiated an NP and been attached to it as a determiner, the rules to parse adjectives are activated. The rule ADJECTIVE has the pattern [adj], and matches the word *falling. Falling* is then attached and disambiguated as an adjective. Recognition of *falling* as a verb does not occur. As there are no more adjectives, ROBIE will activate the rules to parse the headnoun. (ROBIE's grammar assumes that all words between the first noun and the head noun of an NP are nouns; see section 2.6.) The rule NOUN with the pattern [noun] will match on the word *block,* and it will be attached as a noun. Hence *block* will also be disambiguated without the verb use being considered by ROBIE.

Other ambiguities inside the noun phrase will be handled in a similar way. This approach will usually cover the situation of singular head nouns, verb/adjective ambiguity and many other pre-nominal ambiguities. This works because the noun phrase has a very strict word order. When an ambiguous word is found, only one of its meanings will be appropriate to the word order of the noun phrase at that point. This approach can be thought of as an extension of the basic approach of the Harvard Predictive Analyzer (Kuno 1965).

This strategy will also often disambiguate main verbs. For example, consider the following sentences:

(2) Tom hit Mary.
(3) Tom will hit Mary.
(4) The will gave the money to Mary.

In (2), *hit* is the main verb. In the dictionary, *hit* is also defined as a noun, (as in card playing). The parser will attach *Tom* as the subject of the sentence and then activate the rules for the main verb. Since *hit* has the feature "verb", it will match that rule and be attached and disambiguated as a verb. Again, other possible parts of speech are not considered.

The word *will* could be a noun or a modal as sentences (3) and (4) demonstrate. In (3), *will* cannot be part of the headnoun with *Tom,* so the NP will be finished as above. The rules for the auxiliary will then be activated and the word *will* then matches the pattern [modal] and is attached to the AUX.

In (4), the word *will* is used as a noun. Since it follows the determiner, the rules for nouns will be activated. The word *will* then matches the pattern [noun] and attaches to the NP as a noun.

The same approach will also disambiguate *stop* and *run* in the following sentence. Since *stop* is sentence initial and can be a tenseless verb, the rule IMPERATIVE will match, and it will be disambiguated as a verb. The word *run,* which can be a noun or a verb, will be handled as *will* in (4).

(5) Stop the run.

2.5 THE WORD *TO*

Now let us consider a more difficult example, the word *to. To* is defined as an auxiliary verb and a preposition in ROBIE, as illustrated by these sentences:

(6) I want to kiss you.
(7) I will go to the show with you.

In (6), *to* is the infinitive auxiliary, while in (7) *to* is a preposition. This analysis is based on that of Marcus (1980:118). Our two buffer cell lookahead is sufficient to disambiguate these examples.

The buffer patterns for the above sentences are:

[*to*&tenseless] → embedded VP
[*to*&ngstart] → PP

By looking at the following word, *to* can be disambiguated. In (7), the word *the* cannot be a tenseless verb, so the first pattern does not match. In (6), the second buff-

er does not have the feature "ngstart", so the rule doesn't match.

However, the above patterns will accept ungrammatical sentences. To reject ungrammatical sentences, we can use verb subcategorisation as a supplement to the above rules. One cannot say:

(8) *I want to the school with you.
(9) *I will hit to wash you.

In English, only certain verbs can take infinitive complements. *To* can only be used as an auxiliary verb starting a VP when the verb can take an infinitive complement. Hence, by activating the rules to handle the VP usage only when the infinitive is allowed, the problem is partly reduced. Also by classifying the verb for PPs with the preposition *to,* the problem is simplified. This is merely taking advantage of subcategorisation in verb phrases. Taking advantage of this subcategorisation greatly reduces, but does not eliminate, the possible conflict.

We have seen what to do if the verb will only accept a toPP or a VP. The final difficult situation arises whenever the following three conditions are true:

- the verb will accept a toPP and a toVP,
- the item in the second buffer has the features "tenseless" and "ngstart" and,
- the toPP is a required modifier of the verb.

Although this situation rarely arises, the above rule will make the wrong decision if the ambiguous word is being used as a noun. In this situation, ROBIE will make the wrong decision, and has no capability to better decide. By default, the principles of Right Association and Minimal Attachment apply as discussed in Frazier and Fodor (1978).

A free text analysis done on a cover story in TIME magazine (1978) resulted in 55 occurrences of the word *to.* The two rules mentioned above in conjunction with verb subcategorisation gave the correct interpretation of all of these. These rules were also checked on the MECHO corpus (Milne 1983) and the ASHOK corpus (Martin, Church, and Patil 1981). There were no violations of these rules in either of these.

2.6 ADJECTIVE/NOUN AND NOUN/NOUN AMBIGUITY

Adjective/noun ambiguity is beyond the present scope of this research and is handled in a simple-minded way. If the word following the ambiguous adjective/noun word can be a noun, then the ambiguous word is used as an adjective. In other words, all conflicts are resolved in favour of the adjective usage. This problem arises in these examples:

(10) The plane is inclined at an angle of 30 degrees above the *horizontal.*
(11) A block rests on a smooth *horizontal* table.

In (10), *horizontal* is a noun, while in (11), it is an adjective. The above algorithm handles these cases.

This approach takes advantage of the lookahead of the deterministic parser. A word should be used as an adjective if the following word can be an adjective or a noun. However, this approach would fail on examples such as:

(12) The old can get in for half price.
(13) The large student residence blocks my view.

2.7 WHY DO THESE TECHNIQUES WORK?

In this section we have seen many examples of the resolution of ambiguity. To handle these examples, we merely constructed a compound lexical entry for each word, composed of the features of each part of speech the word could be and allowed the pattern matching to perform the disambiguation. This technique has been used by Winograd (1972). Why does this work so well?

English has a fairly strict structural order for all the examples presented here. Because of this, in each example we have seen, the use of the word as a different part of speech would be ungrammatical. Although these techniques have been used for non-deterministic parsers, their effectiveness has not been investigated for a deterministic parser.

Most ambiguities are not recognised by people because only one of the alternatives is grammatical. In many situations, when fixed constituent structure is taken into account, other uses of an ambiguous word are not possible and probably not even recognised. Since fixed constituent structure rules out most alternatives, we have been able to handle the examples in this paper without any special mechanisms. In the introduction to this paper, it was stated that a clean and simple method of handling ambiguity was desired. I feel that this goal has been met for these examples.

3 THE ROLE OF AGREEMENT IN HANDLING AMBIGUITY

Using the simple techniques presented in the last sections, we can handle many cases of part-of-speech ambiguity, but there are many examples we cannot resolve. For example, the second of each pair of sentences below would be disambiguated incorrectly.

(14) I know that boy is bad.
(15) I know that boys are bad.
(16) What boy did it?
(17) What boys do is not my business.
(18) The trash can be smelly.
(19) The trash can was smelly.

Many people wonder what role person/number codes and the relatively rigid constituent structure in the verb group play in English. In this section, we will explore their role by attempting to answer the question, "What use is the fixed structure of the verb group and person/number codes?"

3.1 UNGRAMMATICAL SENTENCES

Before we proceed, let us look at an assumption Marcus made in his parser, that it would be given only grammatical sentences. This assumption makes life easy for someone writing a grammar, since there is no need to worry about grammatical checking. Hence no provision was made for ungrammatical sentences and the original parser accepted such examples as:

(20) *A blocks are red.
(21) *The boy hit the girl the boy the girl.
(22) *Are the boy run?

This simplification causes no problems in most sentences, but can lead to trouble in more difficult examples. If the parser's grammar is loosely formulated because it assumes it will be given grammatical examples only, then ungrammatical sentences may be accepted. If the syntactic analysis accepts ungrammatical sentences as grammatical, then it is making an error. Using grammatical constraints actually helps parsing efficiency and disambiguation. In the next sections we look at the consequences of this assumption as well as those of rejecting ungrammatical sentences.

3.2 SUBJECT/VERB AGREEMENT

We know that the verb group has a complicated but relatively fixed constituent structure. Although verbals have many forms, they must be mixed in a certain rigid order. We also know that the first finite verbal element must agree with the subject in person and number. That is, one cannot say:

(23) *The boy are run.
(24) *The boy will had been run.
(25) *The boys had are red.
etc.

While Marcus's parser enforced these observations to some extent, he did not follow them throughout his parser. We want to enforce this agreement throughout ROBIE. Checking the finite or main verb, to be sure that it agrees in number with the subject, will lead to the rejection of the above examples. This was done by adding the agreement requirement into the pattern for each relevant rule as will be explained later.

Buffers 1 and 2 must agree before a rule relating the subject and verb can match. This check looks at the number code of the NP and the person/number code of the verb and checks whether they agree. The routine for subject/verb agreement is very general and is used by all the subject/verb rules. The routine can only check the grammatical features of the buffers.

3.3 MARCUS'S DIAGNOSTICS

Marcus (1980) did handle some part-of-speech ambiguities. The words *to, what, which, that,* and *have* could all be used as several parts of speech. For each of these words he also used a **Diagnostic** rule. These Diagnostic rules matched when the word they were to diagnose arrived in the first buffer position and the appropriate packets were active. Each diagnostic would examine the features of the three buffers cells and the contents of the Active Node Stack. Once the diagnostic decided which part of speech the word was being used as, it either added the appropriate features, or explicitly ran a grammar rule. Marcus did not give each word a compound lexical entry as we have done here.

Most of the grammar rules in his parser were simple and elegant, but the diagnostics tended to be very complex and contained many conditionals. In some cases they also seemed rather ad hoc and did not meet the goal of a simple, elegant method of handling ambiguity.

For example, consider the THAT-DIAGNOSTIC:

```
[that][np] → in the Packet CPOOL      (Clause pool of rules)
"If   there is no determiner of second
      and there is not a qp of second
      and the nbar of 2nd is none of massn,npl
      and 2nd is not-modifiable
then  attach as det
else  if c is nbar
      then label 1st pronoun, relative pronoun
      else label 1st complementiser."
                                    (Marcus 1980:291)
```

Notice that if the word *that* were to be used as a determiner, then it would be attached after the NP was built! This is his primary rule for disambiguating the word *that.* Marcus's parser also had three other rules to handle different cases.

It seems that these rules did not "elegantly capture generalisations" as did the rest of his parser. I consider these rules undesirable and feel that they should be corrected to comply with my criteria for simple and elegant techniques in resolving ambiguity. I wanted a method that used no special mechanism, or routine, other than that needed to parse grammatical sentences. These diagnostics are certainly special mechanisms and do not meet this goal. Can we cover the same examples in a more simple and principled way?

In this section, we look at each of these diagnostics in turn and show how they have been replaced in the newer model. We also look at a few other examples of ambiguity which Marcus did not handle, but are related to our discussion here.

3.4 HANDLING THE WORD *TO*

The handling of *to* by Marcus's diagnostic can be replaced by the method outlined in Section 2.5. This method was motivated to handle grammatical sentences and meets our criterion for a simple approach.

3.5 HANDLING *WHAT* AND *WHICH*

For both *what* and *which,* the ambiguity lies between a relative pronoun and a determiner. The following examples show various uses of both words:

(26) Which boy wants a fish?	**det**
(27) Which boys want fish?	**det**
(28) The river which I saw has many fish.	**rel. pron.**
(29) What boy wants a fish?	**det**
(30) What boys want is fish.	**rel. pron.**

There is some debate about the part of speech to be assigned the word *which.* Some linguists consider it to be a quantifier (Chomsky 1965), while others consider it to be a determiner (Akmajian and Heny 1975, Chapter 8). We shall adopt the determiner analysis, making the problems for *what* and *which* similar.

To determine the correct part of speech for these two words, Marcus (1980:286) used the following diagnostics:

```
[which] → in the packet CPOOL
"If   the NP above Current Node is not modified
      then label 1st pronoun, relative pronoun
      else label 1st quant,ngstart,ns,wh,npl."

[what][t] → in the packet NPOOL
"If   2nd is ngstart and 2nd is not det
      then label 1st det,ns,npl,n3p,wh;
           activate parse__det
      else label 1st pronoun,relpron,wh."
```

These diagnostics would make the word in question a relative pronoun if it occurred after a headnoun, or a determiner if the word occurred at the start of a possible noun phrase.

If we follow the approach in the last section, and give each word a compound lexical entry composed of the determiner and relative pronoun features, we find that these words are always made determiners unless they occur immediately after a headnoun. In other words, the *which* examples are all parsed correctly, but (30) is parsed incorrectly. This happens because the determiner rule will always try to match before the rule for WH questions can take effect. This simple step gives the correct analysis if the ambiguous word is to be a determiner, but will still err on (30).

The rule to parse a relative pronoun and start a relative clause is active only after the headnoun has been found. At this time, the rule for determiners is not active. Therefore, if the word *what* or *which* is present after a headnoun, the only rule that can match is the rule to use it as a relative pronoun, and it will be used as a relative pronoun. We have resolved the simple case of *what* as a relative pronoun using only the simple techniques of the last section. For these sentences

(31) What block is red?
(32) Which boy hit her?
(33) Which is the right one?

ROBIE produces the correct analysis, but still errs on (30). This error is because *what* is being used as a relative pronoun but does not follow a headnoun. Without any additional changes to the parser, we get two things. Firstly, if the word occurs after the headnoun, then the NP-COMPLETE packet rules are active, and it will be a relative pronoun. In fact, since relative clauses can occur only after the end of an NP, this correctly resolves the relative pronoun uses. If the word occurs at the start of an NP, then it will be made a determiner.

This approach has exactly the same effect and coverage as did Marcus's diagnostics, but we have not needed any special rules to implement it. It will now provide the correct interpretation for *which,* but will make some errors for the word *what.* Marcus's **what-diagnostic** will treat *what* as a determiner whenever the item in the second buffer could start a NP. This is usually correct, but *what* will be treated as a determiner in all of the following:

(34) What boys want is fish.
(35) What blocks the road?
(36) What climbs trees?
(37) What boys did you see?
(38) What blocks are in the road?
(39) What climbs did you do?

In this paper, we are adopting the following analysis for WH clefts such as (34). The initial WH word, *what* is a relative pronoun and attached as the WH-COMP of the subject S node. The subject is the phrase *What boys want.* The main verb of the sentence is *is* and the complement *fish.* The exact details are not important, only that the word *what* or *which* is a not determiner at the start of a WH cleft.

In sentences (34-36), the word *what* is not used as a determiner. In the analysis we are using, it is a relative pronoun and is used as the WH-COMP for the S. In sentences (37-39), the word *what* is used as a determiner. Marcus (1980:286) admits that this diagnostic produces the incorrect result in this case. His diagnostic will make *what* a determiner in all of these examples, as will my analysis.

One can also see that each of the above pairs is a pair of potential garden path sentences. For each pair, the two buffers contain the same words. Hence our two-buffer lookahead is not sufficient to choose the correct usage of the word *what.* Using only two or three buffers, there is no way to make *what* a relative pronoun when the headnoun is plural but a determiner when it is singular for all arbitrary sentences.

With regard to the Semantic Checking Hypothesis (Milne 1982) then, it is suggested that this decision is based on non-syntactic information. I believe that intonation is critical in these examples. Unfortunately there is insufficient experimental evidence to determine for certain whether this is true. Finally, the problem of *what* and *which* as sentence initials, with no noun in the second buffer seems to arise very rarely. I have found no examples of this problem in free text analysis.

The current parser (ROBIE) cannot obtain the extra information provided by intonation to help resolve this case. As a result it follows Marcus's diagnostic and makes *what* a determiner in each of the above cases.

This is because *what* is defined as a determiner that can agree with either a singular noun or a plural noun, as it was in Marcus's parser.

3.6 HANDLING *THAT*

In ROBIE, *that* is defined as a singular determiner, a pronoun, a relative pronoun, and a complementiser. Marcus had four diagnostics to handle the word *that*. We have seen one of these at the start of this section. In this sub-section we see how these four diagnostics can be replaced in a simple way. Let us consider how to handle the uses of *that* one at a time.

Firstly, as a determiner. The following sentences illustrate the problem in identifying this usage.

(40) I know that boy should do it.
(41) I know that boys should do it.

Marcus assumed that PARSIFAL would be given only grammatical sentences to parse. If determiner/number agreement is not given to a parser, then it will, incorrectly, make *that* a determiner in (41), producing the wrong analysis. The way to prevent this is to enforce number agreement in the rule DETERMINER by insisting that the determiner agree with the noun in number. The determiner usage will be grammatical only when the headnoun has the same number. If we make this a condition for the rule to match, then *that* will not be made a determiner in (41) and ROBIE will get the correct parse.

For this case, the agreement check would make sure that one of the following patterns match:

[det,ns] [noun,ns]
[det,npl] [noun,npl]

The above two cases are handled properly because number agreement blocks the interpretation of the (41) as a determiner. This approach leads to the correct preference, when there is an ambiguity and accounts for the difficulty in (42) versus (43):

(42) That deer ate everything in my garden surprised me.
(43) That deer ate everything in my garden last night.

The second experiment in Milne (1983), showed that (42) is a garden path sentence, while (43) is not. In both sentences, it is believed the subject uses the word *that* as a determiner. *Deer* is both singular and plural, so it fits the above rule. In (42), *that* must be used as a complementiser to make the sentence grammatical. The approach outlined above will use *that* as a determiner in an ambiguous case such as this.

These two simple techniques, word order and agreement, are sufficient to handle all the examples we have just presented. In addition, free text analysis has shown no violations to this approach (Milne 1983). These techniques provide the same coverage as Marcus's diagnostic, with the added bonus that the determiner is attached before the NP is built.

That can only be a complementiser when a *that S-* is expected. Hence the rules using *that* to start an embedded sentence are only activated when the verb has the feature THAT-COMP. The rules in THAT-COMP will fire when *that* is followed by something that can start an NP. This ensures that the S- will have a subject and means that *that* will be taken as a pronoun in the following sentences:

(44) I know that hit Mary.
(45) I know that will be true.

but it will be taken as a complementiser in these sentences:

(46) I know that boys are mean.
(47) I know that Tom will hit Mary.

It seems that, unless the S- has a subject, the pronoun use of *that* is preferred. Otherwise one would have a complementiser followed by a trace, rather than a unmarked complementiser, followed by a pronoun. This rule provides more complete coverage than Marcus's diagnostic since it examines the second buffer.

The rule to handle pronouns in general is of low priority and will only fire after all other uses have failed to match. *That* is treated in the same way.

That will be identified as a relative pronoun only if it occurs after a headnoun and the packet NP-COMPLETE is active. This situation will be handled in the same manner as the usual relative clause rules and will then cover:

(48) I know the boy that you saw.
(49) I know the boy that hit you.

The most difficult case for *that* is when the verb is subcategorised:

V NP S-

That is, it can take an NP subject, followed by a *that* S-. For these examples, ROBIE may have to decide if the series of words following *that* is a relative clause or an embedded sentence.

In the following sentences, the lookahead would have to be more than three buffers. (Brackets indicate words in the buffers. The last word is the disambiguating word.)

(50) I told the girl [that][the][boy] hit the *story*
(51) I told the girl [that][the][boy] will kiss *her*

It can be seen that in these sentences the disambiguating word is outside our three buffers. How do people handle these, and what should our parser do? In Milne (1983) it was shown that when the syntax could not resolve the ambiguity with its two-buffer lookahead, the decision of which interpretation to use might be made using non-syntactic information. It was also stated that if context can affect the interpretation of the sentence, then non-syntactic information is being used to select the

interpretation. The reader can experiment for himself and see that context does affect the interpretation of these sentences. Therefore it is predicted that non-syntactic information is being used to interpret these sentences, and that this problem should be resolved not on a semantic basis but on a non-syntactic one.

This explains why some of these examples cause difficulty and others do not. The psychological evidence from cases using *that* is scant, and I feel no conclusions can be reached here. My theory predicts that context will strongly affect these examples and, if they are strongly biased to the incorrect reading, a garden path should result.

One well-known example in this area is (52):

(52) I told the girl that I liked the story.
(53) I told the girl whom I liked the story.
(54) I told the girl the story that I liked.

These examples were tested in Milne (1983). The results suggested that (52) was read faster than the other two examples. Many of the subjects were questioned informally after the experiment about their interpretation of the sentence. All reported only one meaning; the S-reading. None of the subjects said that they noticed the relative clause reading, hence the result. The experiment however, was not designed formally to distinguish these.

To handle the examples we have seen in this section, Marcus had four diagnostics, one of which was very complicated. I have just shown how to handle all four cases of *that* without any special rules, merely substituting enforced agreement and rejecting ungrammatical sentences.

3.7 HANDLING THE WORD *HAVE*

Let us now look at the elimination of Marcus's HAVE-DIAGNOSTIC in relation to the use of agreement we have been discussing in this section. The problem with *have* is illustrated by the following sentences:

(55) Have the students take the exam.
(56) Have the students taken the exam?

In these, we must decide if *have* is an auxiliary verb or a main verb and whether the sentence is a yes-no question or an imperative. The sentences have the same initial string until the final morpheme on *take*. To handle this case, Marcus (1980:211) used this rule:

"RULE HAVE-DIAG PRIORITY:5 IN SS-START
[have,tenseless][np][t] →
If 2nd is ns,n3p or 3rd is tenseless
then run imperative next else
If 3rd is not verb
then run yes-no-question next
else if not sure, assume it's a y/n-q and run yes-no-question next"."

This rule seems to be necessary in order to distinguish between the question and the imperative. If one tries to ascertain exactly what occurs, the apparent complexity is revealed. Note also that Marcus's rule defaults to a yes-no question twice in this diagnostic. The following sentences illustrate the distinction this rule makes.

(57) Have the boy take the exam.
(58) Have the boy taken the exam.
(59) Have the boys take the exam.
(60) Have the boys taken the exam?

It can be seen that YES-NO QUESTION should run only when the NP following is plural and the verb has "en" (i.e., *taken).* [Only (60) has a plural noun, *the boys,* and the verb *taken.*] This can also be understood as: the sentence is an imperative if the item in the 2nd buffer is not plural and the verb is tenseless. Thus, the first three examples above are Imperatives because either the noun *(boy)* is singular (57 and 58) or the verb is tenseless (59). The second part of the rule takes care of the fact that the third buffer must contain a verb for the imperative, as this would be the main verb of the embedded sentential object.

Let us look more closely at the reason why only (60) is a question. Firstly, if the sentence is a yes-no question, then aux-inversion must occur. When this happens, *Have* will be adjacent to the verb that was in the third buffer. In order for ROBIE to continue, the verb must have an "en" ending, or *have* and the next verb will not agree in aspect. This is the basis for discrimination in the earlier examples (57-60).

Secondly, in (57) and (58), the noun phrases are singular and both sentences are imperatives. Had the sentence been a yes-no question, *have* would need to agree with the subject, which must then be plural.

Hence, in effect, Marcus's rule checks for number agreement between the subject and verb, and checks that the fixed order of the verb group is obeyed. Let us now look at other situations where this is necessary.

PARSIFAL would accept the following ungrammatical strings:

(61) *Are the boy running?
(62) *Has the boys run?
(63) *Has the boy kissing?
(64) *Has the boy kiss?

For a yes-no question, the inverted auxiliary must agree with the verb after it has been inverted. To stop these ungrammatical constructions, we must enforce verb agreement. The pattern for the rule YES-NO QUESTION should be:

[auxverb][np][verb], agree(auxverb,verb),agree(verb,np).

This constraint enforces agreement of the verb and auxiliary verb and the subject and verb. Again this check is based only on the linguistic features of the buffers.

Such a constraint effectively blocks the ungrammatical constructions. (The parser will fail if the auxiliary has been inverted, since the auxiliary will not be parsed.) Also the subject NP must agree with the auxiliary verb, so we can also add "agree(auxverb,np)" to the rule, as

we did with the HAVE-DIAGNOSTIC! So, by correcting the yes-no question rule, the HAVE-DIAGNOSTIC is redundant.

In this section we have seen that Marcus's HAVE-DIAGNOSTIC can be replaced by merely exploiting agreement. It should be pointed out that although this approach has the same coverage as Marcus's diagnostic, it is wrong in some cases. Milne (1983) has a full discussion.

3.8 PLURAL HEAD NOUNS

There is a class of ambiguities that can be resolved merely by enforcing subject/verb agreement. In this section, we see an example from the class of words with the features noun, verb, final-s (plural). If we have two words that can be a plural noun or a singular verb, we can enumerate four cases. Let us look at these possibilities and see that these cases can be disambiguated by simple rules using subject/verb agreement. The following examples illustrate all the possibilities:

(65) The soup pot cover handle screw is red.
(66) The soup pot cover handles screw tightly.
(67) *The soup pot cover handles screws tightly.
{68} The soup pot cover handle screws tightly.
{69} The soup pot cover handle screws are red.

Each of the words *pot, cover, handle,* and *screw* can be either a noun or a verb. The "end of constituent" problem is to find out which word is used as the verb and which words make up the complex headnoun. The possible distributions of the morpheme "s" among two words gives us four cases. We deal with each of these in turn.

Case 1: In (65) each noun is singular. For this case all ambiguous words must be nouns and part of the headnoun. Due to subject/verb agreement, a singular noun must match a 3rd person singular (v3s) verb, i.e, one without the letter "s". This case excludes that possibility since none of the words have an "s" at the end. Hence they must all be nouns.

Case 2: In (66) *handles* is a plural noun and each word before it must be a noun. When a singular noun/verb word follows *handles,* the word *(screw)* must be a verb and *handles* is the last of the headnouns. It is not possible to use *handles* in this situation as a verb, and *screw* as a noun because of subject/verb agreement.

Case 3: The examples in this case have two consecutive plural nouns as in (67), where both words have noun/verb ambiguity. (Do not confuse plural "s" with possessive "'s").

When the first plural is a noun, then the second one can be a verb only if it is part of a different constituent. Examples of this are the following. (Sentences beginning with "?" are considered grammatical but unacceptable to most readers.)

(70) ?The soup pot machine handles screws easily.
(71) The soup pot machine handles screw easily.
(72) Which years do you have costs figures for?
(73) Do you have a count of the number of sales requests and the number of requests filled?

[(72) and (73) are from Martin, Church, and Patil (1981).]

Because there is a non-plural headnoun followed by a plural headnoun, this case is really a subset of Case 4. In general, the problems and issues for Case 4 dominate the resolution of this ambiguity.

Case 4: Sentences (68) and (69) both have the same word initial string until after *screws,* but in (68) *screws* is a verb while in (69) *screws* is part of the headnoun. In this situation, where the final word in a series is plural, each word before it must be a noun. The word itself can be either a noun or a verb, depending on what follows. These can be recognised as a pair of potential garden path sentences, as discussed in Milne (1982). Therefore, this is the case to which the Semantic Checking Hypothesis applies and the predictions of Milne (1982) apply.

In that paper, the idea of **potential garden path sentences** is presented. These are sentences that may or may not lead to a garden path. Each garden path sentence has a partner, which is similar but not a garden path. It is proposed that the decision as to how to resolve the ambiguity that may lead to a garden path should be made by semantics and not by syntax. This theory is called the Semantic Checking Hypothesis. For full details see Milne (1983).

In this section, we have looked at resolving a simple case of noun/verb ambiguity. In order to resolve this ambiguity, it was necessary merely to exploit agreement between the subject and verb in number and person.

Due to number and subject verb agreement, these facts have a linguistic base. They rely on the fact that a final "s" marks a plural noun but a singular verb. If the verb is v3s (verb agrees with a 3rd person, singular noun, as with the "s"), then the subject of the verb must be singular, or else the sentence is ungrammatical. This is why all the words before the v3s word must be nouns. If any of these words were used as a verb, then subject-verb agreement would be violated. This is why (67) is ungrammatical. If the verb is v-3s (agrees with any noun phrase except 3rd person, singular i.e., no "s"), then the subject cannot be singular. (65) has no plural subject and so cannot have a v-3s verb. In (66) *handles* provides a plural subject, so *screw,* which is v-3s, can agree.

3.9 NOUN/MODAL AMBIGUITY

We now consider noun/modal ambiguity as demonstrated by *can* and *will.* Both can be either a noun or a modal (i.e., *could, should, would, can, will, might,* etc.):

(74) The trash can was taken out.
(75) The trash can be taken out.
(76) The paper will was destroyed.
(77) The paper will be destroyed.

Each of these words is entered in the dictionary both as a noun and a modal. Due to agreement requirements, the modal/noun word can only be grammatically used as a modal if the word following it is a tenseless verb, i.e., the pattern:

[modal][tenseless] → modal usage

applies. Handling noun/modal ambiguity can be quite easy; when the noun modal word appears in the first buffer one merely has to look at the contents of the second buffer to see if it contains a tenseless verb. This can be complicated, though, if the auxiliary is inverted or the sentence is an imperative. The following examples show how this can arise:

(78) Let the paper will be read.
(79) Will the paper can be re-used?

In sentence (78) the fragment *Let the paper* implies that *will* can only be used as a noun, as the sentence already has one tensed verb. In the parser, the noun/modal word is first encountered inside the NP packets and the parser must decide whether to use the word as part of the headnoun or to leave it in the buffer to be used as a modal verb. These rules do not know whether a verb has been found previously. Hence, not all information from the sentence is used. If all the information is available at the time the noun/modal ambiguity is being resolved, these sentences would be unambiguous and people would have no trouble reading them.

Subjects were asked to read the above examples in the second experiment presented in Milne (1982). The results showed convincingly that they are potential garden paths. Many naive readers had considerably more difficulty with them than with their more straightforward counterparts. This was predicted for reasons explained below.

This result seems surprising. If the subjects used all information available at the time the noun/modal word was encountered, then they should have had no trouble with these sentences. The fact that these are garden paths indicates that the readers did not use all the information available to them. Notice also that the ambiguity can be reformulated as: "Do we have the end of a noun phrase, or a complex headnoun?"

We have already seen a case where people do not seem to use all the information available to them. In Milne (1983), several end-of-NP problems were presented that could lead to a garden path. In each of these, it was shown that the ambiguity was resolved on the basis of non-syntactic information, without regard to the following words in the sentence. In other words, we saw that the reader did not use all the information available. There is one crucial difference though. In the previous cases, non-syntactic information was used because the syntactic processor with its limited lookahead was sometimes unable to choose the correct alternative. In this case, the information necessary has already been absorbed by the parser.

This suggests that the choice of alternatives is made locally inside the NP parsing rules, without regard to information about the type of sentence being parsed. In other words, the two-buffer pattern applies regardless of the rest of the sentence. This assumes that a noun/modal word followed by a tenseless verb is being used as a modal. This is similar to Fodor, Bever, and Garrett's (1974) old canonical sentoid strategy: a bottom-up analysis that took every N-V combination as a new *S*. Let us look at why this might be true in the parser.

When the parser starts to parse a NP, it creates a new NP node and pushes it to the bottom item of the Active Node Stack. This operation makes the NP node the Current Active Node and parsing of the old Current Active Node is suspended. If the parser is parsing an S node, for example at the start of the sentence, then work on this node will be suspended until the NP node has been completed and dropped into the buffer.

In ROBIE, unlike PARSIFAL, the pattern matcher for the grammar rules is allowed only to inspect the grammatical features of the two buffers. This means that the parser is unable to examine the contents of the Active Node Stack and, hence, the information that a tensed verb has already been found is unavailable to the NP parsing rules. This then suggests that the ambiguity will be resolved on the basis of local information only.

It should be pointed out that although ROBIE does not examine the Active Node Stack, the current packet reflects its contents. For example, if the parser is parsing the major S node, the packet *SS-VP* will be active, but if the parser is parsing an embedded S node, the packet Embedded-S will be active. This information can be considered to provide local context to the parsing rules. This is the same as in PARSIFAL.

This ambiguity is an end-of-NP problem and the choice of alternatives is made on the basis of limited and local information. This suggests that non-syntactic information may be used to resolve the ambiguity. There is one further possibility. The semantic choice mechanism is attempting to find the end of an NP. So far it has asked the question, "Can this item be part of the NP?" However, the end-of-NP problem can be reformulated as, "Is it better to use this as part of the NP, or as the start of the verb group?" It is conceivable that the end of NP mechanism uses *will* as the start of the verb group in the majority of occurrences, hence leading to the apparent modal preference in these examples:

(80) The trash can hit the wall.
(81) The paper will hit the table.

Due to lack of data, it is not clear exactly what people do in this situation and this would seem to provide an interesting area for further investigation.

3.10 WHAT ABOUT *HER*

Another problem is the word *her,* which can be used as a pronoun or as a possessive pronoun. Note that we can say:

(82) Tom kissed her.
(83) Tom kissed her sister.

Clearly in (82) *her* is a pronoun and in (83) *her* is a possessive determiner. When multiple part-of-speech definitions were added to ROBIE and the simple disambiguation method used, ROBIE always made *her* a possessive determiner.

This difficulty arose in Marcus's parser because the rule to start a NP was ordered before the rule to parse a pronoun. These rules were copied directly into ROBIE's grammar. Since the word *her* has both the features "ngstart" and "pronoun", it could match both rules. Unfortunately, as Marcus's rules were stated, it always matched the NP starting rule, and hence was used as a determiner by the parser. This indicates one problem that can arise in the writing of a parser grammar.

To handle possessive determiners, PARSIFAL and ROBIE have a rule with the pattern:

[poss_np]

This rule will match a possessive pronoun after it has been made into an NP. It will also match any possessive NP, such as: *the boy's* or *the boy's mother's.* The rule then adds the feature "determiner" to the NP, making it eligible for the NP starting rule. By degrading the possessive NPs to determiners, both parsers easily handle examples of left branching such as:

(84) The boy's mother's brother is his uncle.

Another problem arose in (82) because the possessive NP rule was not sufficiently constrained. It is possible to use *her* as a determiner only where the next word can be part of a noun phrase with that determiner. To enforce this, the second buffer is checked to be certain that its contents will take the determiner. Using this approach, *her* in (82) would not be converted to a possessive determiner. The rule DETERMINER can run only if the next item will "take a determiner".

This check is made by the syntactic category of the following word, rather than by a specially marked feature. This check could be done by having a list of all the possible categories as the pattern of the second buffer. As an implementation detail, this is in the form of an agreement check, merely to simplify this rule and to show its generality.

The only remaining problem occurs when the verb can take one or more objects and the item after the word *her* can be either the second object, or an NP with *her* as a determiner. For example:

(85) I took her grapes.
(86) He saw her duck.
(87) I gave her food for the dog.

The examples presented above are all examples of global ambiguity, which is discussed in more detail in Milne (1983). In these cases the check of "Will the next word take a determiner?", may or may not lead to the wrong analysis. This problem also interacts with the top-down component of verb phrase parsing and the semantic restrictions presented by it.

The conflict between the determiner and possessive usage can be modelled as a conflict of rule priorities. If the possessive use is preferred, then this rule should match first. Conversely, if the object use is preferred, then the object rule should match first. Any error in reading these examples would be due to one rule having priority over the other, when the reverse should be the case. Finally, notice that with no help from either intonation or context, either analysis is possible. That is, there is not enough information in the sentence to determine a unique interpretation.

We have now shown how to replace all the diagnostics Marcus used. In doing this, we enforced number and verb agreement on the rules before they could run. This was motivated to reject ungrammatical items, rather than for the handling of ambiguity. While there are still a few problems due to global ambiguity, the approach reported here has the same coverage as Marcus's diagnostics, and provides a better explanation of why people have trouble on certain sentences.

4 POSSIBLE USES FOR AGREEMENT IN ENGLISH

In this paper, we have seen several occurrences of ambiguity, for each of which we have found a parallel situation that could lead to acceptance of ungrammatical sentences by ROBIE. We then used person/number codes or the fixed structure of the verb group to block these unacceptable readings. Most of our ambiguity problems were also handled by this method. Although this has been used before with non-deterministic parsers, it was not obvious that it would provide enough information to enable deterministic parsing.

Once person/number codes are taken into account, the number of potential ambiguous readings is dramatically reduced. In many cases, only one of the ambiguous possibilities was grammatical. It should be noted that there are a few difficult cases which we have not had time to describe in this paper; these are discussed in detail in Milne (1983).

Marcus had a few rules to resolve part of speech ambiguity, but they were ad hoc. We have seen that we can replace these rules very simply by merely exploiting agreement.

In the introduction, it was stated that handling lexical ambiguity was a major test for deterministic parsing. In this paper we have seen that many cases of ambiguity can be resolved in a simple way. This is possible because of the constraints imposed by number agreement and word order. In fact, many cases of the seemingly difficult

problem of lexical ambiguity turn out to be easily resolved in a deterministic parser, since the deterministic parser uses more information to make decisions.

References

Akmajian, A. and Heny, F. 1975 *An Introduction to the Principles of Transformational Syntax.* MIT Press, Cambridge, Massachusetts.

Chomsky, Noam 1965 *Aspects of the Theory of Syntax.* MIT Press, Cambridge, Massachusetts.

Dewar, H.; Bratley, P.; and Thorne, J. 1969 A Program for the Syntactic Analysis of English Sentences. *Communications of the ACM* 12(8).

Fodor, Jerry; Bever, T.; and Garrett, M. 1974 *The Psychology of Language.* McGraw-Hill, New York, New York.

Fodor, Janet and Frazier, Lynn 1978 The Sausage Machine: A New Two-Stage Parsing Mode. *Cognition* 6: 291-325.

Kuno, S. 1965 The Predictive Analyzer and a Path Elimination Technique. *Communications of the ACM* 8(10).

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

Martin, William; Church, K; and Patil, R. 1981 Preliminary Analysis of a Breadth-First Parsing Algorithm: Theoretical and Experimental Results. MIT AI Lab. Presented at Modeling Human Parsing Strategies Symposium, Austin, Texas.

Milne, Robert 1982 Predicting Garden Path Sentences, *Cognitive Science* 6: 349-373.

Milne, Robert 1983 Resolving Lexical Ambiguity in a Deterministic Parser. D.Phil. Dissertation, University of Edinburgh, Edinburgh, Scotland.

TIME 9 January 1978 Good Ole Burt; Cool-eyed Clint.

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