WORD PREDICTION FOR DISABLED USERS:

APPLYING NATURAL LANGUAGE PROCESSING TO ENHANCE COMMUNICATION

By

Julie A. Van Dyke

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Honors Bachelor of Arts in Cognitive Studies.

June 1991

WORD PREDICTION FOR DISABLED USERS:

APPLYING NATURAL LANGUAGE PROCESSING TO ENHANCE COMMUNICATION

By

Julie A. Van Dyke

Approved:____
Kathleen F. McCoy, Ph.D.
Professor in charge of thesis on behalf of the
Advisory Committee

Approved:____
William J. Frawley, Ph.D.
Committee member from the Linguistics Department

Approved:____
Roberta M. Golinkoff, Ph.D.
Committee member from the University Honors Program

Approved:____
Robert F. Brown, Ph.D.
Director, University Honors Program

ACKNOWLEDGEMENTS

Two years ago I began this work as a nifty way to spend a summer. Within a few weeks, I realized just how complex the English language was and that my Allen & Greenough's New Latin Grammar didn't explain grammar as completely as I thought it did. Suddenly I had embarked on a project that would forever change the way I looked at language. Here, I would like to acknowledge those individuals and organizations without whose guidance and support this magnum opus could not have been possible.

First and foremost, I am grateful to my family: especially Mom, Dad, and Grandma. Ever since third grade, when I camped out in the living room for a week while building a bookworm house out of dozens of hand-made miniature books, you have graciously encouraged me through all my projects that inevitably turn out to be bigger than life.

This particular project, however, was different from all those because it allowed me to grow while it grew; all the while learning, seeing, and accomplishing a great deal. For this I owe a world of thanks to Kathy McCoy, who, although never willing to admit it, is a great student of Plato the Greek. Countless times I walked into her office thinking that I had not much to talk about and left 2 hours later with a whole new direction for the project that she somehow teased out of my proliferous brain.

I also owe unbounded thanks and respect to Bill Frawley, who buoyantly announced in class one day, “You’ll have to bear with me here, I get excited about this stuff.”. That single quote and this man’s artistry in the classroom showed me how exciting linguistics could be and what going to school was all about.

I must also express my sincere thanks to Pat Demasco and the A.I. DuPont Institute’s natural language lab. They provided me the motivation for this project, the freedom to shape it as I pleased, and made a lowly undergraduate feel at home. In particular, I am grateful to Linda Suri for keeping me on my toes with impossible syntax questions and for her help with the final revisions of this work.

In addition to guidance from the Computer Science Department, I also had the pleasure of being adopted by the University of Delaware Linguistics Department. I received invaluable help, challenges, and inspiration from many people there. Specifically, I would like to thank Tom Ernst, Peter Cole, Jim Lantolf, Roberta Golinkoff, and Gaby Hermon for their various roles in making this work a success. This is one undergraduate for whom your department has had a significant impact, and I hope one of many in the years to come.

Lastly I need to recognize the faculty and staff of the University Honors Program for their incorrigible support, and for giving me a home away from home.

This work was done in conjunction with the A.I. DuPont Institute’s Applied Science and Engineering Laboratories and was partially funded by the University Honors Program.

TABLE OF CONTENTS

FIGURES

TABLES

ABSTRACT

Disorders such as Cerebral Palsy and Lou Gehrig's disease produce severe physical disabilities that make normal communication impossible. This project addresses this problem by developing a syntactic prediction system. Other communication aids have previously been developed using abbreviation and prediction to enhance communication, but these have had limited success. Abbreviation systems allow the user to type pre-determined, shortened word-forms which the computer is responsible for expanding. Prediction systems attempt to predict the user's next keystrokes based on statistical data.

I have combined natural language processing techniques and popular syntax theories to devise a prediction system that, unlike these previous systems, models the syntax rules that specify how words can be combined. This allows the syntactic predictor to make rule-based, linguistic determinations about what words can follow those already processed. It can be used with flexible abbreviation systems to eliminate possible expansions for personalized abbreviations. The syntactic predictor could also be used with other devices to reduce the effort required of the user by predicting what word forms he or she is likely to type next. In modelling linguistic knowledge, this system provides a more natural solution to the communication problem than many systems currently in use.

Chapter 1

AUGMENTATIVE COMMUNICATION

In this chapter I will outline the current state of augmentative communication which is the field providing the motivation and most immediate application of this project. I will characterize the potential users for this system and then discuss the goals that I believe this project achieves in relation to other work in this field.

1.1 Background

Rehabilitation engineering endeavors to integrate technology into vocational, educational, and independent living settings in order to increase the independence of persons with physical or sensory disabilities. There are numerous sub-areas in this field: robotics research for artificial limbs, production of sensory aids like hearing aids, developing physical therapy techniques, and devising marketing strategies to promote this technology. The project described here derives its motivation from the sub-field of Rehabilitation Engineering called Augmentative and Alternative Communication (AAC). This work is an example of how natural language processing techniques and modern syntax theories can be used to improve the communication devices currently available to disabled users.

Communication devices are technological interventions between the disabled user and the world he or she hopes to communicate in. The history of their development includes intervention strategies that were not able to take the needs and desires of the user into consideration because of technological limitations. Often they placed a cognitive or physical burden on the user because they were not intuitive or were complicated to operate. I will outline some of these systems in the following sections, but it is encouraging to keep in mind that the field has reached a new consciousness. The primary consideration has changed from finding any way possible for these people to communicate, to finding the communication aid that is best suited to each individual. The emphasis is now on developing intervention devices which feature easy training, flexible use, and allow a reasonable communication rate.

The project I will present in the following chapters is a product of these new concerns. It will contribute to the development of an intervention device that preserves the user's ability to use language freely while maintaining speed and ease.

1.2 The User

The typical user for the system developed here is cognitively intact and therefore has the mental capability and desire to use language the same way a non-disabled individual would. The user's disability affects his or her motor capability and muscular control in a way that produces limited dexterity. These users are typically non-speaking, and have difficulty typing, writing, or even controlling a joy-stick to select letters. In the worst case the user is limited to using a single-switch interface which makes communication very slow.

Two types of disorders that typically produce this condition are developmental, such as Cerebral Palsy, or degenerative, like Lou Gehrig's disease. Cerebral Palsy (CP) is diagnosed at infancy in children whose normal muscular control is deficient (Griffith, 1985). The child usually exhibits unusual body postures, purposeless body movements, and poor coordination and balance. Although some children with CP may suffer from mental retardations, many have a high intelligence despite their muscular disabilities. I am targeting these individuals as possible users for the technology I am describing here.

The clinical name for Lou Gehrig's disease is Amyotrophic Lateral Sclerosis (ALS). This disorder afflicts adults late in their lives, meaning that there is no previous language impairment or cognitive disability (Griffith, 1985) to hinder communication. Patients suffer from muscle twitching and weakness, beginning in the hands and spreading to the arms and legs, or from the stiffening of muscle groups usually in the extremities. They will often lose control of muscles that perform swallowing and communicative functions.

Some stroke victims could also benefit from this technology; however, often they will have more severe linguistic impairments which make using this system inappropriate. Whatever the ailment, if the user can be characterized as linguistically, or cognitively, intact but with deficient motor skills, the device I have developed has the potential to facilitate their communication.

1.3 Available AAC Devices

In this section I will outline some of the AAC systems that are currently available for users with the characteristics I have identified. There are still a lot of non-electronic communication aids available for disabled users; however, I am concerned here only with electronic ones because they are compatible with this project. Typically these use the user's motor capability (albeit limited) to compose messages via an electronic system. Because his or her motor capability is limited, the system must require minimal effort from the user.

Many severely disabled individuals find a single switch device useful for communication. The switch is used to access letters on a one-to-one basis as he or she composes a message. One of the first of these devices to be developed was the Tufts Interactive Communicator (TIC), which consisted of a small mechanical box and switch (Foulds, 1976). Thirty-two possible characters are offered to the user on an electronic grid appearing on the face of the box and each row is scanned until the user selects the one containing the desired item with the switch. Then the machine scans each column in that row until the user selects the desired letter or word. As the user selects each character, it is typed onto a paper strip printer which the user can use for communication purposes.

This sort of interface is common because it can be effectively used by many clients. It is only one of several selection methods available, all of which attempt to provide the user with a manageable way to communicate in spite of his physical limitations. Clearly an important issue in developing these devices is the speed with which the user can compose a message. It is easy to imagine that communication with a single switch device like the TIC is a slow and laborious process, and in fact average communication rates are around 2-10 words per minute (Foulds, 1980). Compare this to non-disabled typing speeds of 60-70 words per minute or to speaking speeds which are easily twice that, and the extent of the communication deficiency for these users is clear. It is a deficiency resulting only from the technology available to them, because as I have described previously, these user's cognitive and linguistic abilities are intact. The user is just without the muscular control necessary to use their skills in a normal fashion.

Because of this problem, AAC research has focussed on developing strategies to increase the communication rate possible with these devices. For scanning systems like the TIC, this has principally meant devising variations in the order that letter-characters are offered to the user. Knowledge about the frequency of letter usage is used to rearrange the letters on the TIC display so that the more frequent ones are scanned before the less popular ones. For instance, in a row-column scan, the most frequent letters of the alphabet such as "S" or "T" could be placed in the upper left-hand corner of the display. This way the scan will cross these letters first and thereby avoid scanning through unlikely choices like "V" or "Q" most of the time. This technique was able to produce a 30% improvement over alphabetically-ordered letter displays (Foulds, 1976).

Another improvement on this scanning technique is a type of letter prediction in which the system uses more sophisticated frequency data to control the scanning. These statistics take the form of n-gram statistics which tell what n number of letters are likely to occur together, such as "str" or "ing". With this system, if the user has already indicated an "s" the system refers to its statistics to identify the six letters that are likely to follow the “s”, like for instance “t”, “p”, “r”, “e”, “i”, or “h”. These are immediately highlighted in sequence so that the user has the opportunity to select them before the normal row-column scanning process is resumed. By anticipating letter selection in this way, the communication rate has been improved by up to 50% (Foulds, 1976).[1]

The scanning technique I have been describing can also work at the word level, and the ordering improvements in this case are based on what words are the most frequent. In contrast to TIC, which uses hard-wired letter grids, Meta4 is a software-based communication device that uses static word pages containing the most common words (Miller, 1990). Instead of having to spell out each word, the user navigates through the pages using the single switch. The system’s first page might contain letter intervals such as “AA-AL” and “AL-AZ” and the scanning passes through these intervals until the user chooses the one containing the word he wants to use. Then the display changes to show a page containing vocabulary words that he can choose from using the same scanning technique. The words included on these pages are a vocabulary set, called a “book”, that can be tailored for each individual user. It is possible for users to have several books to choose from and in this way a large amount of vocabulary can be made available in a way that does not require the user to spell out every word letter by letter. In the case that the word that the user wants to use is not in any of the vocabulary books the system has there is a spelling page the user can select from the initial display. This works just like the TIC system, using the scanning technique to allow the user to spell out the new vocabulary

1. This is based on knowing 3 prior characters so the statistics would take the form of quadgrams such as “stri”.

word.

Dynamic communication devices, like Meta4 which has a changing display, can also be improved using prediction techniques. Word prediction systems try to determine the next word based on what has already been entered into the system. One example of prediction used this way is the PAL system developed at the University of Dundee (Swiffin, et al. 1987). This system uses frequency statistics to determine the word that is most likely to follow what has already been entered. The statistics are gathered from large samples of text, wherein the frequency of each word is tabulated and included in the dictionary entry of that word. When the user types a letter, the system displays the five most frequent words beginning with that letter in a special scanning window. The user can choose one of these words or type another letter. With each keystroke the frequency statistics are checked and possible completions for the word are offered to the user in the scanning window. In this way the system attempts to predict what the word is before the user has typed it out entirely. The number of keystrokes required of the user is reduced because the system completes the word as soon as the user indicates that the right one has been found. Those who developed PAL claim that they have been able to obtain a rate reduction of 50% based on a dictionary of 1000 words, wherein each word has its own frequency data.

An effort was made to improve PAL even further by including syntactic information to their statistical data. A probability matrix for category pairs was produced to constrain the word prediction according to the syntactic class of the previous word. For example, after an adjective is entered, the probability of a specific noun as its successor is computed by multiplying the pre-computed probability of an adjective-noun pair and the noun's occurrence frequency. This was done for each noun in the dictionary and the most probable ones were offered to the user. With this use of syntax, PAL's developers were able to reduce the number of keystrokes necessary to generate sentences by an additional .5-2% (Swiffin et al., 1987).[2]

While the PAL system has been successful at reducing the number of keystrokes required of the user, it is important to note that this reduction has been found using a fairly limited dictionary of 1000 words. Because of the way PAL uses syntax, dictionaries of a much larger size are likely to severely degrade the performance because it will take longer to calculate the probability of each word using the word-pair statistics and consequently it will take longer to determine the five most probable words.

This raises a problem that is common to all these systems I have been discussing: they depend on statistics, rather than the rule-based linguistic information that humans actually use when they communicate. Because of this the system is only as effective as the statistics are accurate and complete. As with the PAL frequency counts, statistics are typically collected over a large texts often derived from newspapers and published reading materials. This means they are liable to be skewed by the subject matter of the text. For example, the Brown Corpus of American English, which is a text of approximately one million words and one that is often used for deriving statistics for these systems,

2. Depending on the text and the number of predictions offered. These data are for dictionaries of between 700-1000 words.

represents words like “eggs” and “bunny” and “Easter” as being common words in everyday language use. This is a result of the time of year that the corpus was compiled, not of actual facts about English usage. This problem can be solved to some extent by using statistics derived from the user’s own language use; however, the same problem can occur with these texts because a user does not always talk about the same topics and so the word statistics could change depending on his topic of conversation. In school frequently used words might be “homework” or “teacher” but when a child is playing these words will be the least likely words that he will use. A problem with statistically-based systems also arises when novel words are used. The system has no statistics for these words and so despite the statistical information in his or her AAC device, the user will still have to completely spell out words.

Non-stochastic strategies for improving communication rates have centered around abbreviation systems. Instead of spelling out words letter-by-letter, the user is able to use an abbreviation. He or she can indicate fewer letters with the scanning device and the system will assume responsibility for expanding the abbreviations. A major problem with the abbreviation systems now available is that the user has to memorize specific abbreviations for words in order for the system to be helpful. This arises because a computational system that can only handle a one-to-one correspondence between a word and its abbreviation. Thus, the system may require the word “work” be abbreviated “wrk” in order to differentiate it from the word “wake”, which might be abbreviated “wk”. The user will consider this need when he is trying to communicate and so he may not think to use “wrk” instead of the more easily constructed “wk”. Because of these predetermined abbreviations, the user must undergo specialized training to learn the system's abbreviations before he or she can even start using the system.

In addition, these abbreviation systems and scanning devices assume that the user knows how to spell the word he is trying to use. A communication device called Minspeak (Baker, 1985) was an attempt to alleviate this problem as well as those associated with memorizing pre-determined abbreviations. This system used a keyboard of multi-meaning icons together with keys for morphological and rudimentary syntactic information to create sentences. For example the client might use the key sequence [boy-image] + [noun-key] + [smiley face-image] + [verb-key] + [book-image] + [building-image] + [noun-key] + [declarative-sentence key] to compose the sentence "Boy like school." Once the sentence is composed, the client will press a "speak" key and the computer will speak the phrase the user created. This use of images allows the abbreviations to be semantically meaningful to the user and presumably easier for him or her to remember.

Minspeak has proven useful for many members of the disabled community, but it has also been problematic for some because it still requires the user to understand and/or memorize the associations between the images and the English words. When the system was conceived, it was intended that each user should make up his or her own icons corresponding to the words he or she used most. Clinically this proved to be an enormous and impossible task for the clinicians whose responsibility it was to set up the vocabulary for each user. As a result, the system is used with the scheme of its creator and this may not be intuitively clear to some users. Minspeak, therefore, suffers from the same disadvantage as other abbreviation systems in that the user has to undergo an extensive training period before he or she can use the device. Even after this period the user may not fully grasp the semantic justifications underlying particular icons so that he or she is never able to fully exploit the system's power.

A less revolutionary technique for improving abbreviation systems has been attempted, with flexible abbreviation systems such as the "Word Compansion" project described in (Demasco et al., 1989) and in (Stum et al., 1991). These systems attempt to automate the methods humans use for creating abbreviations so that the computer can associate more than one abbreviation with a particular word. This means the computer will be able to handle "wk" as an abbreviation for any of "work", "wake", "walk", "wok", etc. The user can be freer with his abbreviations and the system's success does not rely on how well the user remembers the abbreviation the computer knows for the word he or she desires. The very benefit of accepting a single abbreviation for many words is a problem because the computer is faced with disambiguating the proper word from a list of many candidates. Since this is a computational burden rather than a burden on the user, it is a more desirable solution.

In order to expand the abbreviations, the system assumes the letters in the abbreviation are in the same order that they occur in the word. This makes expansion similar to a matching task: it assumes variables between the known letters and tries to match this form to the more than 5000 words in the dictionary the system currently uses.

The problem with this is that 5000 words is a small dictionary for the requirements of everyday communication so it is desirable to have a larger dictionary. With this dictionary the number of matches for a given abbreviation may be very high and this means the user could still have to expend a considerable amount of time and effort to find the desired abbreviation among many possibilities.

The task as always, is to improve the behavior of the flexible abbreviation device, and in this case that will be done by reducing the number of candidates for the expansion of any user-created abbreviation. I have discussed some ways communication devices have been improved using statistics and prediction techniques; however, it has been shown that all of these strategies have their limitations. One obvious solution which has not been fully developed in the field of Augmentative Communication is to exploit linguistic knowledge. A priori this seems to be the best solution because it uses exactly the knowledge the user draws from when he constructs sentences. Only PAL, with its category pairs, attempted to use syntax. This was very rudimentary; however, as it only considered statistical word pairs, and therefore missed many of the generalities in the language. For example consider the sentence "The man saw the dog eating his food." If the system can only look at word-pairs it will not know that the word-pair "the man" is likely to be followed by a verb whereas the word-pair "his food" is not because of its overall position in the sentence. A more refined model of syntax which considers the entire sentence preceding the current word could increase accuracy and yield more efficient communication using the augmentative communication devices I have been describing (Yang et al., 90).

1.4 Linguistic Improvements

In this project I develop a prediction technique that exploits the linguistic rules of syntax.This will allow capturing the generalities of language rather than artifacts of the data the statistics were taken from. Using rule-based linguistic information rather than word distributions better models the way humans join words to construct sentences that have particular syntactic structures underlying them. Sentences are created by using a grammar of word categories and production rules about how the categories can be joined. If these rules are incorporated into a communication device, the users will be limited only by the information they use anyway as part of their language faculty. He or she is not limited by the computer's statistics and therefore is allowed the utmost flexibility in using language.

One of few existing systems which exploit linguistic knowledge in the way I am proposing was developed at the A.I. DuPont Institute's Applied Science and Engineering Laboratories. This is a sentence "compansion" system (McCoy et al., 1990) that takes an abbreviated, or compressed, sentence like "John walk dog" and expands it into the sentence "John walks the dog." This allows the user to produce grammatically well-formed sentences without the extra keystrokes necessary to indicate plurality and spell out non-content words like "the".

This system uses a semantic parser that determines the role of each input word (i.e., verb, noun, etc.) and assigns theta roles to each noun in the sentence. The theta roles are determined according to a "frame" which is analogous to the theta-grid that

Government and Binding Syntax posits as part of the lexical specification of verbs (Sells, 1985). For example the frame for the verb "study" indicates that the AGENT must be human and that it can take a THEME (i.e., an abstract or physical object) and a LOCATION (i.e., a physical place). The frame further specifies that the AGENT is required and both the THEME and LOCATION are optional. Using this semantic information about the input words, the system can fill out the sentence with the appropriate inflections and non-content words. For example the compressed sentence "John study red house" will be interpreted as either "John studies at the red house" or "John studies the red house" depending on how the semantic roles are filled. In this way the system uses semantic knowledge about the individual words the user enters to produce well-formed sentences that still require minimal effort from the user. This system uses semantics to solve the computational problem inherent in allowing compansion (i.e., COMPressed input exPANSION).

This expansion problem at the sentence level is similar to the problem at the word level which I described in the flexible abbreviation systems: the user abbreviates the word and the computer must expand it. The syntactic prediction system I have developed can be used to aid word expansion in the same way that semantics was used for sentence expansion. I propose that the number of expansions of an abbreviation can be greatly reduced by considering the syntactic categories of the expansions in relation to the syntactic structure of the words the system has already processed. For example, if the user has entered the partial sentence "The boys" and the next word abbreviation is "ht", instead of offering the user a long list like "hit, hits, hot, hat, hate, hates, height, hunt, hunts, hurts, hurt, hut," the user will only be offered the plural verbs in this list because it will know that nouns and adjectives are not appropriate once the head noun of a noun phrase has been identified.[3] In this case, the user will only need to choose from the words "hit, hate, hunt, hurt" rather than a list of twelve choices.

Notice that this also increases the user's communication rate because he or she has fewer words to scan through before finding the desired word. Only the words syntactically appropriate to the context will be offered as possible expansions for the user's arbitrary abbreviations. With this strategy implemented in tandem with the Word Compansion system, the goal of achieving speedy communication while making minimal demands on the user will be within reach.

In addition to its usefulness with flexible abbreviation expansion, the system I have developed is a prediction system that could be used to improve other communication devices by determining the syntactic form of the word that is likely to follow what the user has already entered. For example in a dynamic system like Meta4, if the user has already entered a noun, and the user chooses the interval ST-SZ, the system could go directly to a page containing only verbs that begin with those letters. This would further increase communication rate because the user will have fewer words to look through before finding the desired one.

Thus, by modeling syntactic knowledge in the computer, I can produce a system that can improve existing communication devices. The improvement provided is a more

3. For this example I am assuming noun-noun modification is not allowed.

natural one for the user because it comes from the information humans use anyway when they communicate. It is not an ad hoc solution to the communication problems these people face, it is a solution motivated by the nature of the problem: an inability to use language in a “natural” unconstrained way. If we can make the machine use language the way a human does, then rather than being hindered by the technology the user’s disabilities force him to use, both machine and human can cooperate to enhance the disabled person’s communication.

Chapter 2

THE PREDICTOR

This chapter contains the implementation details of the syntactic predictor I have built, including a sample of its operation. I discuss the underlying concepts borrowed from Natural Language Processing as well as the computational formalism used.

2.1 Concepts and Definitions

Syntax is the component of human language processing that describes phrase and sentence structure. It can be described with a finite set of rules specifying how word categories can be combined in well-formed sentences. Natural Language Processing (NLP) characterizes these rules as rewrite rules with the form X --> Y. These rules are meant to transform the expression on the left side of the arrow, X in this case, to the form on the right side, represented by Y (Allen, 1987). To illustrate these rules, which are also called “phrase structure rules”, consider the sample context-free grammar below:

(1)

Rule number	Left-Hand Side		Right-Hand Side
1	S	--->	NP VP
2	NP	--->	N
3	NP	--->	DET N
4	NP	--->	NP PP
5	PP	--->	PREP NP
6	VP	--->	V
7	VP	--->	V NP
8	VP	--->	V NP PP
9	DET	--->	a\an\the\some...
10	N	--->	John\man\dog...
11	V	--->	walk\hit\open...
12	PREP	--->	with\of\in...

Parsing is the process of applying rules like these to a sentence to break it down into its component parts. The result is a “parse tree” that shows the syntactic categories and functional relationships between the constituents in the sentence. Applying the rules above to the sentence

(2) The man walked the dog.

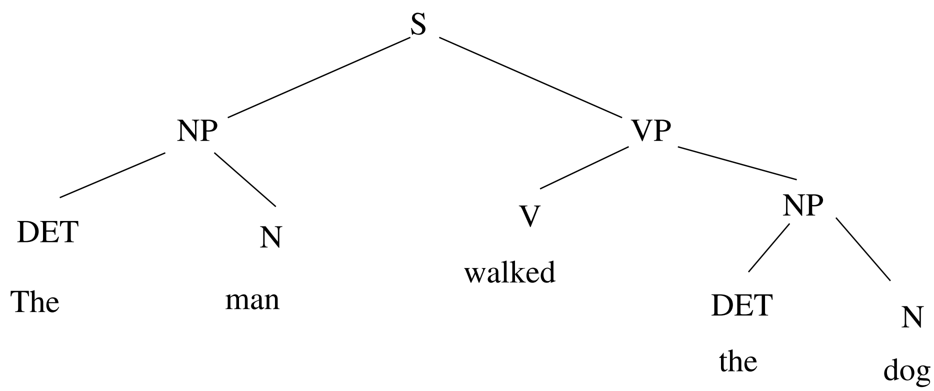
gives the following parse tree, or "parse", shown in computational notation:

(3) (S (NP (DET the) (N man))

(VP (V walked) (NP (DET the) (N dog))))

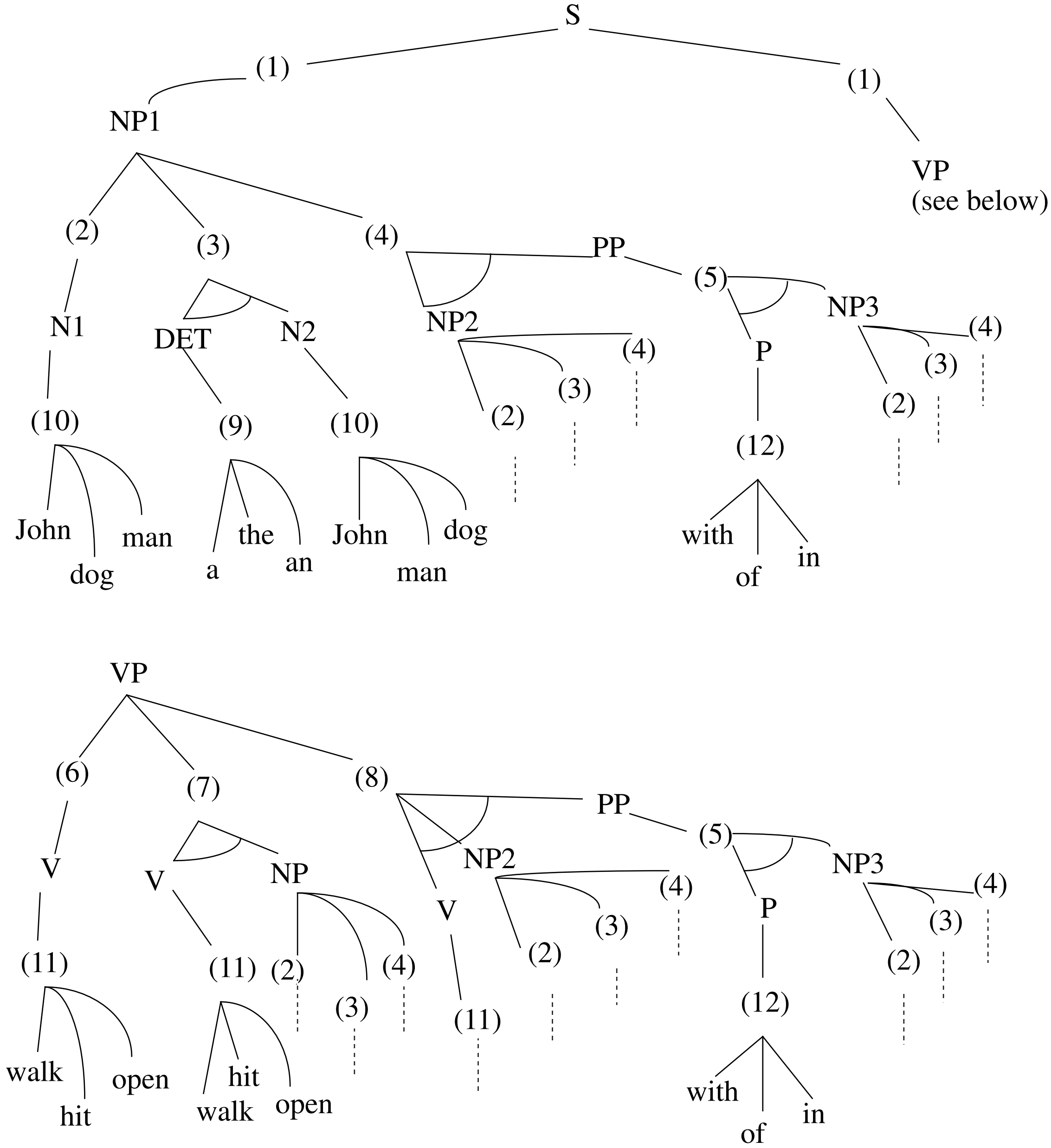
A noun phrase is labeled "NP", verb phrases are "VP", and each word is given an appropriate category label such as "DET", "N", or "V". This structure represents the more commonly known tree structure below:

(4)



To generate this parse, the computer needs to search all the possible combinations of the rules in grammar (1). This becomes complicated because grammars normally have different ways of expanding constituents, as in the case of the NP's and VP's in (1). Any of these combinations might be possible, so the computer must try them all until it finds the right one. The final parse ends up being a subset of the overall search space. The search space itself can be very large; the search space for the small grammar in (1) looks like Figure 2.1 below. Vertical dots are used to indicate where parts of the search space have been left out.

Figure 2.1 Search Space for a Context Free Grammar



This is only part of the complete search space. In reality it is infinitely deep because of recursive elements like the NP. Each time an NP occurs it can be broken into three different groups of constituents, here represented by nodes 2, 3, and 4, which correlate with the grammar rules having the same numbers. Since rule 4 (as well as 5, 7, and 8) has NP's as part of its structure the search tree can never be completely expanded.

I have drawn this tree in a manner that illustrates the different rules that can expand a constituent. When the daughters of a rule number are tied together with an arc it means that both these elements must be present in the input for the rule to be successful. The grammar rules 9-12 provide the primitives, which are the lexical items in the sentence, for the tree. In parsing, if the actual item in the sentence is one of the lexical items in rules 9-12, then that rule can be considered complete.

In order to find the parse tree of (4) the computer traverses the search space in Figure 2.1. The method it uses might be a "top-down, depth-first" method, where it starts at the top S node and tries to make its way down to the primitives, where it can check them against the actual input.[1] Processing will start at the top and go down the tree, starting on the left side, as far it can. When it reaches a primitive or a point where no rules apply to the input, the processing backs up and goes down another branch of the tree.

For example, consider the search space in Figure 2.1 while parsing to get the structure in (4). The computer first uses rule 1 to expand S into NP1. Then it tries rule 2 and finds that it needs an N. Since the first word of the sentence (2) is "The" this path fails and the processing backs up to the NP1. Next it tries rule 3 and finds that it must complete a DET and this succeeds with the word "the". Because the DET is connected via an arc to

1. I am describing a top-down method because the computational formalism I am using in this project as well as the predictor I have constructed from this formalism work in a top-down way. It is also possible to traverse the search space in a number of other ways, including a bottom-up method.

a N path, the processor must complete both paths before rule 3 will be successful. It therefore backs up to try look for the N in the other part of rule 3. This succeeds with "man" and so rule 3 is completed and the processing returns to NP1. The next branch of the tree is that generated by rule 4. In this case, the input is the word "walked" and the computer will try this rule, fail, and processing will continue to the VP. Here again there are 3 possible rules for expanding the rest of the sentence. Taking the left-most branch gives a single verb generated by rule 6. This would work with the input "walked" and so it is taken. But now the rest of the sentence is "the dog" and the processing will continue trying rules 7 and 8 to account for that noun phrase. When it reaches the end and it finds no rule that includes it because both 7 and 8 expect verbs next, the computer will back up and choose not to take 6 (undoing what it has already done). It will take rule 7 instead and since this is composed of a V and an NP, this rule will succeed. Since there is no more input the processing will stop; the parse in (4) having been found. In this way the computer tries each path in the search space, beginning from the left-most one, until a successful traversal is found

2.2 ATN Formalism

From an NLP standpoint, a parser is a machine that, when given a grammar written according to its specifications, will carry out the search process I described in the previous section. The Augmented Transition Network (ATN) (Woods, 1969) is one such machine that has been very successful in NLP implementations and which is widely available (Bates, 1978). This machine actually has more computational power than what is needed for the processing described above, but here I will discuss the ATN only as it can be used for the parsing natural language grammars.

The parser itself is implemented to perform a top-down, left-to-right parse using the method I described in the previous section. Note that a grammar of English is intended to account for all and only the grammatical English sentences. Consequently, there is a problem with the grammar given in (1) because it is unrestricted and would allow sentences like (6):

(6) a. *Boy walk dog.

b. *John hit.

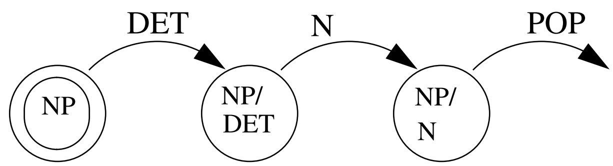
c. *The girl cried the man.

These parses could be eliminated by adding tests of particular features of each word to the word categories specified in the rules of (1). For example sentence (6c) could be ruled out by a test checking to see if the main verb has the feature “intransitive”. If the verb is intransitive, it cannot have a noun phrase following it as a transitive verb requiring a direct object would. Processing would be carried out in this case such that before the ATN executes a grammar rule, it executes any tests that are specified within the grammar to check if that particular rule is applicable to the input. Aside from eliminating ungrammatical sentences, tests can be used to make the processing more efficient. For example assume, as was the case in the example in the previous section, that the next word of input is “walked” and the rule that the parser is considering is one for expanding an NP. The test might say “if the input word can begin an NP, then execute this rule, otherwise proceed to the next rule." In this way the grammar itself can restrict the amount of searching the parser needs to do and rule out some bad sentences without complicating the phrase structure rules.

In addition to declaring whether or not particular sentences are grammatical, it would be more useful to build up a parse tree like that seen in (4). The ATN does this by attaching actions to the grammar rules. When a rule is executed, these actions are performed to assign the input its grammatical category and build the tree structure. The actions might also test the structures that have already been built and on that basis interrupt a particular rule's execution. For example, if the main verb does not agree with its subject or does not have the inflection a preceding auxiliary calls for, the path the parser is following will fail and the processing will be forced to back up and try another parse. The structures that are built are stored in a set of "registers", which are place-holders for information which can be used later in tests, actions, or constructing additional structure. For example if a sentence is determined to be passive, actions could be constructed to assign the structure held in the "subject" register to the object register, thus making room for the new subject.

The ATN represents actual grammar rules, such as those in (1), in the form of networks which show a transition from one state to the next. This transition is analogous to each step towards completing the rule; a phrase structure rule like "NP --> DET N" has a transition between NP and DET and one between DET an N. The transitions are depicted as arcs in a network as follows:

(7)



The double-circle around the NP node identifies it as the start state of the network. The labels of the intermediate states show what constituents of the rule have been completed (i.e., NP/DET means an determiner has been processed already in the NP network). The final state is the one having the arc labeled “POP”, which is an indication that the rule is complete.

The formalism provides several different ways of describing the transitions between parts of a phrase structure rule. The most useful is the CAT arc, which checks to see if the category specified by the phrase structure rule matches that of the input. The CAT arc might have the following form, given in LISP notation:

(8) (CAT DET t (setr DET *) (to NP/DET))

In this arc, “CAT” is a label telling the parser what sort of processing is necessary, in this case to check the category of the input word. The symbol “DET”, for determiner, specifies the category that the phrase structure rule is looking for. The “t” is in the position where a test like those I described earlier would go. Since the act of checking the category will tell whether or not the transition can be made, no test is necessary and a dummy test allows processing to continue. The “(setr DET *)” is the action that assigns the word of input, represented by *, the name of its syntactic category. The “(to NP/DET)” tells the parser where to go next, in this case to the state after the DET transition has been made.

Other transitions are programmed in the same way with appropriate tests and actions. The main difference is in the first label signifying what kind of processing the parser needs to do in order for the transition to be completed. One of the most important kinds of transitions, or arcs, is the “PUSH” arc. This accounts for the recurrence of constituents like the NP in many rules. It signals the parser that it needs to temporarily leave the present rule and process the rules for expanding the NP. These are represented by separate networks, and because they can be used over and over again, the size of the grammar is small in relation to the size of the sentence structure it can account for. When the NP is completed, the transition has been completed and the parser returns to the original network to continue working on a particular phrase structure rule.

Other kinds of arcs include WRD arcs, which allow a phrase structure rule to specify that a particular word be in the sentence; JUMP arcs, which allow for processing to proceed to a different state without any actions or checking being done; MEM arcs, which require the word of input to be one of a particular set of words; and POP arcs which signal that a network is complete and provide for building larger structures out of the constituents most recently processed.

A special kind of arc called the VIR arc helps to account for movement in English. There are certain English sentences, such as wh-questions, in which a constituent moves from its original position in the sentence into a new position at surface structure. The object of the sentence might be moved out of object position and replaced with a wh-word, as in the sentence

(9) What did John eat?

The underlying structure of sentence (9) is

(10) John did eat what.

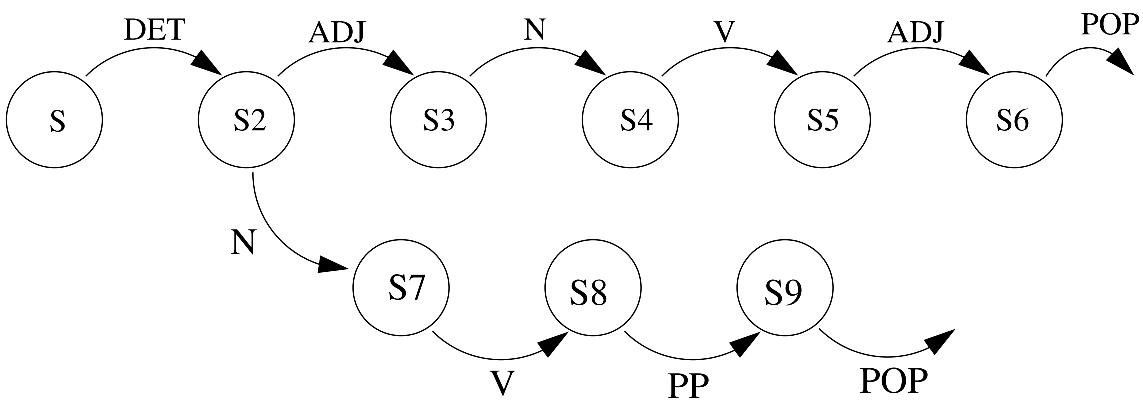
The ATN processes (9) by using a “hold-list” and VIR arcs to return the moved constituent to its original position. When the computer encounters the wh-word “what” it is processed as an NP and put on the hold-list. A VIR arc occurs in the grammar at the place where the constituent has moved from (i.e., in object position of sentence (10)). When a VIR arc is encountered in the grammar, instead of looking for a constituent in the string of input, the NP is taken from the hold-list to satisfy the phrase structure rules.

With this mechanism, the ATN can undo transformations that have occurred to derive the surface structure it is processing. The VIR arc is used to signify the positions from which a constituent could have originated and the “hold-list” allows the parser to wait before assigning a constituent its position in the final sentence structure. This process is used whenever sentences are left with “holes” after movement has occurred, as is the case with relative clauses as well as the wh-movement explained here.

2.3 The Prediction Problem

As I mentioned previously, the ATN has proven very useful for problems in natural language processing. It is not useful for prediction; however, because it follows one parse at a time and backtracks if it reaches a dead-end. To do prediction, the system must take a partial sentence and return the features and category of the next input word. But because the ATN does not follow all parses at once, it does not have access to all possible next words of input. This is especially clear when words with category ambiguity are used in sentences. For example, consider the simplified grammar network below:

(11)



If the system only has the partial sentence "the" and the word "gold" is entered, the parser does not know whether "gold" is an adjective or a noun. The ATN, parser as I have described it, would choose one path down the network and follow it. Consequently it may not adequately predict the category of the word that follows "gold": with network (11) it will predict a noun to be next as if the sentence were "the gold ring is beautiful." It is just as likely that a verb could be next; however, such as if the sentence were "the gold is in the bank."

As a result of this "one-at-a-time" method of parsing, the ATN may be forced into continual back-tracking each time a word is entered. With each path change, possible predictions would be unaccounted for because the computer would only be following one path at a time. If the computer took “gold” to be an adjective, at that point in the processing it cannot predict that the next word could be a verb as well as a noun. This means that the prediction would be incomplete in a significant number of cases because a typical grammar requires a large number of paths to account for the many structures in English. In addition, this would make the processing much slower and therefore it would be difficult to use this system for spontaneous communication.

The way to solve this problem is to change the way the ATN parses so that it will complete all possible parses at once. This is done by making the processing do a top-down, breadth-first traversal of the search space and in this way the ATN simulates a parallel processing mechanism that will generate all possible parses at once. Now the ATN analyses “gold” as a noun in one parse and as an “adjective” in another. When the next word is entered, it may eliminate one of these interpretations, or else the ATN will continue both parses until the entire sentence has been entered. Either way, the parser is able to know at any point in the sentence what type of word could be next, because it is holding all possible structures for the words entered thus far.

2.4 Implementation

The parser I have built solves the prediction problem by traversing the search space in Figure 2.1 in a breadth-first, rather than top-down manner. This means that it completes the first transition in each phrase structure rule before going deeper in the tree. Essentially, the depth-first parser needs to only maintain one parse at a time; however, this breadth-first parser constantly maintains all the partial parses so that at each point it knows all the categories that could be used to complete a grammatical sentence. When a new word is given, each parse incorporates it into the structure it has been building. If the word cannot be included in a parse, that parse is eliminated from further consideration. This means the processing is done in a non-deterministic fashion, and therefore complete predictions can be made because the computer has not committed itself to a particular parse that may turn out to be different from what the user intended. This also means that when the entire sentence has been entered, the parser may have built more than one structure for a particular sequence of words.

Because of this exhaustive analysis of the search space, the parser can also account for different structures underlying the same words. For example, consider the sentence:

(12) The man told the woman that he loved the story.

The user could have meant either that the indirect object is “the woman that he loved” and the object be “the story” or that the indirect object be “the woman” and the object be “that he loved the story.” The predictor will output both these structures so that they could easily be analyzed further by a semantic or pragmatic processor that may eliminate one interpretation based on the context the user has been building.

This predictor has been implemented in SUN Common LISP. There is also an early implementation in Franz Lisp. It is intended as a component in a more complex communication system and as such, there has been little attention paid to the user interface. Presently the system is activated with the command “predict” and a partial sentence given as its argument. The system goes as far as it can with that partial sentence and then goes into a “break package” where the user can decide between two methods of proceeding. The first method allows the next word in the sentence to be entered. It incorporates that word into the partial parses already created by the system and then reenters the break package. At each point when a parse is completed, the system prints out that parse tree. These parses are not final analyses, as they can still be given additional words that will be incorporated into them. The system halts only when there is no possible way of continuing the parse given the input it already has. In this case the predictor returns “nil.”

The second method for continuing from the break package is to enter a series of words (e.g., possible abbreviation expansions) and the system returns those words which could possibly be next, given the partial sentence it has already processed. Once the eliminations have been made, the break package resumes and the user is again given the two choices for proceeding until he signals that he wants to quit. An example of this operation is given in the following section.

The grammar that the predictor uses to create and judge grammaticality is described in more detail in Chapter 3. Recall that part of the function of the grammar arcs is to carry out tests of particular features on the input words to determine if it is efficient to carry out a particular rule. These features are encoded in the dictionary entries for each word that the computer knows. The dictionary and the features within it are described in more detail in Chapter 4 on the lexicon.

In order to help with adding words that the user wants to use but that the computer does not have in its dictionary there is an auxiliary package used at run-time to check each word entered against those in the dictionary. When the computer finds a word it does not know, this package allows that word to be added automatically in the dictionary. The package gives the user directions for entering the appropriate features for each word to ensure that the dictionary entry is of the form the grammar expects (cf. Chapter 4).

2.4 Performance

What follows is a brief demonstration of the way the predictor works. This is actual output from the system as the sentence "The gold key on the table opened the door easily" is entered in parts. Note that it is possible to enter more than a single word at once, as was the case with the phrases "on the table" and "opened the door" in the example.

During the course of entering this sentence, various lists of words were given to the system for it to choose those that could grammatically follow the part of the sentence that the system has already processed. These lists are meant to show how the predictor can eliminate in appropriate input, and do not reflect what might be logically entered in a discourse. When a complete sentence is formed with any of the words entered via method 1 or with words in a list of possibilities, the sentence parse incorporating that word is printed.

(predict ‘(the))
The parser has gone as far as it can with this sentence.
You can choose to proceed in two ways:

Method 1 allows you to enter the next word to be completed.
The computer will then advance the parse as far as it can with this new word and then bring you back to this point.

Method 2 allows you to enter a list of words and the computer checks these words and tells you which of them could be used to advance the parse a further step. At that point you are returned to this point and you can choose to take one of those words or to check another list.

The sentence parsed so far is:
(THE)

Now if you would like to use Method 1 type 1.
If you would like to use Method 2 type 2.
If you would like to quit this program type 0.
Enter your selection now:
1

Enter the next word: (gold)

This word was not in the dictionary
GOLD
Do you wish to enter it? Choose (y) or (n)
(y)

It will be entered now.

Enter the categories for this word.
Possibilities: DET PRO PREP ADV N ADJ V CONJ

Enter the category in list form (i.e., enter ‘(N)’). If a word belongs to more than one category make this a list also (i.e., ‘(DET PRO)’)

(N ADJ)

Enter the features for the word that have values.
Your choices are:
FOR N: NUMBER
FOR ADJ: NUMBER ZONE

Remember that all values for different interpretations of the word must be entered in the same dictionary entry.
Enter these features with their appropriate arguments in list form.
For example ‘(TAKES (SGCT) NUMBER (SG))’.
(NUMBER (SG) ZONE (3))

Enter the toggled features for the word.

Possibilities are:
FOR ADJ: QUANT, CENTRAL-DET, PRE-DET, PRED
FOR N: CARDINAL, MASS, COUNT, PROPER, PRE-DET, MENSURAL

Remember to include all that apply to this word in all of its meanings.
Enter them in list form like '(ART CENTRAL-DET)'
(MASS)

Enter the root of the current word.
Only VERBS require roots, but other inflected words may have them.
If this word has not root, enter 'nil'.

nil

The parser has gone as far as it can with this sentence.
You can choose to proceed in two ways:

Method 1 allows you to enter the next word to be completed.
The computer will then advance the parse as far as it can with this
new word and then bring you back to this point.

Method 2 allows you to enter a list of words and the computer
checks these words and tells you which of them could be used to
advance the parse a further step. At that point you are
returned to this point and you can choose to take one of those
words or to check another list.

The sentence parsed so far is:
(THE GOLD)

Now if you would like to use Method 1 type 1.
If you would like to use Method 2 type 2.
If you would like to quit this program type 0.

Enter your selection now:

2

Enter the list of possible next words:
(key keys on in open the a)[2]

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP
        (SPEC
         (DP (SPEC NIL)
           (DBAR (HEAD THE)
              (COMP
```

2. This list is meant to demonstrate the predictor's ability to eliminate ungrammatical word forms based on the sentence parsed so far. When actually applied with a flexible abbreviation system, this list would look more like (hit, hat, hot, hunt, hurt, hate, etc.).

```
                    (NP (SPEC NIL)
                        (NBAR (HEAD GOLD)
                              (NU (SG))))))))
          (IBAR (HEAD (AGR 3PLPRES))
                (COMP (VP (SPEC NIL)
                          (VBAR (HEAD OPEN))))))))))
```

The grammatical possibilities are: (KEY KEYS ON IN OPEN)

The parser has gone as far as it can with this sentence.
You can choose to proceed in two ways:

Method 1 allows you to enter the next word to be completed.
The computer will then advance the parse as far as it can with this new word and then bring you back to this point.

Method 2 allows you to enter a list of words and the computer checks these words and tells you which of them could be used to advance the parse a further step. At that point you are returned to this point and you can choose to take one of those words or to check another list.

The sentence parsed so far is:
(THE GOLD)

Now if you would like to use Method 1 type 1.
If you would like to use Method 2 type 2.
If you would like to quit this program type 0.
Enter your selection now:
1

Enter the next word: (key)

The parser has gone as far as it can with this sentence.
You can choose to proceed in two ways:
Method 1 allows you to enter the next word to be completed.
The computer will then advance the parse as far as it can with this new word and then bring you back to this point.

Method 2 allows you to enter a list of words and the computer checks these words and tells you which of them could be used to

advance the parse a further step. At that point you are returned to this point and you can choose to take one of those words or to check another list.

The sentence parsed so far is:
(THE GOLD KEY)

Now if you would like to use Method 1 type 1.
If you would like to use Method 2 type 2.
If you would like to quit this program type 0.
Enter your selection now:
2

Enter the list of possible next words:
(key keys open opened on in a the)

```
(DCL
 (CP (SPEC NIL)
    (CBAR (HEAD NIL)
        (COMP
          (IP
            (SPEC
              (DP (SPEC NIL)
                 (DBAR (HEAD THE)
                     (COMP
                       (NP (SPEC NIL)
                          (NBAR
                            (ATTRIB
                              (DEGP (SPEC NIL)
                                  (DEGBAR (HEAD NIL)
                                        (COMP
                                          (AP (SPEC NIL)
                                             (ABAR (HEAD GOLD)))))))
                            (HEAD KEY)
                            (NU (SG))))))))
            (IBAR (HEAD (AGR 3SGPAST))
```

The grammatical possibilities are: (OPENED ON IN)

The parser has gone as far as it can with this sentence.
You can choose to proceed in two ways:

Method 1 allows you to enter the next word to be completed.
The computer will then advance the parse as far as it can with this new word and then bring you back to this point.

Method 2 allows you to enter a list of words and the computer checks these words and tells you which of them could be used to advance the parse a further step. At that point you are returned to this point and you can choose to take one of those words or to check another list.

The sentence parsed so far is:
(THE GOLD KEY)

Now if you would like to use Method 1 type 1.
If you would like to use Method 2 type 2.
If you would like to quit this program type 0.
Enter your selection now:
1

Enter the next word: (on the table)

The parser has gone as far as it can with this sentence.
You can choose to proceed in two ways:

Method 1 allows you to enter the next word to be completed.
The computer will then advance the parse as far as it can with this
new word and then bring you back to this point.

Method 2 allows you to enter a list of words and the computer
checks these words and tells you which of them could be used to
advance the parse a further step. At that point you are
returned to this point and you can choose to take one of those
words or to check another list.

The sentence parsed so far is:
(THE GOLD KEY ON THE TABLE)

Now if you would like to use Method 1 type 1.
If you would like to use Method 2 type 2.
If you would like to quit this program type 0.
Enter your selection now:
2

Enter the list of possible next words:
(open opened in a the)

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP
        (SPEC
         (DP (SPEC NIL)
           (DBAR (HEAD THE)
              (COMP
               (NP (SPEC NIL)
                 (NBAR
                  (ATTRIB
                   (DEGP (SPEC NIL)
                      (DEGBAR (HEAD NIL)
                          (COMP
                           (AP (SPEC NIL)
                             (ABAR (HEAD GOLD)))))))
                  (HEAD KEY)
                  (NU (SG))
                  (COMP
                   (PP (SPEC NIL)
                      (PBAR (HEAD ON)
                        (COMP
                         (DP (SPEC NIL)
                           (DBAR (HEAD THE)
                              (COMP
                               (NP (SPEC NIL)
                                 (NBAR
                                  (HEAD TABLE)
                                  (NU (SG)))))))))))))))))
```

```
        (IBAR (HEAD (AGR 3SGPAST))
           (COMP (VP (SPEC NIL)

(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP
        (SPEC
         (DP (SPEC NIL)
           (DBAR (HEAD THE)
              (COMP
               (NP (SPEC NIL)
                 (NBAR
                  (ATTRIB
                   (DEGP (SPEC NIL)
                      (DEGBAR (HEAD NIL)
                          (COMP
                           (AP (SPEC NIL)
                             (ABAR (HEAD GOLD)))))))
                  (HEAD KEY)
                  (NU (SG))
                  (ADJUNCT
                   (PP (SPEC NIL)
                      (PBAR (HEAD ON)
                        (COMP
                         (DP (SPEC NIL)
                           (DBAR (HEAD THE)
                              (COMP
                               (NP (SPEC NIL)
                                 (NBAR
                                  (HEAD TABLE)
                                  (NU (SG)))))))))))))))))
        (IBAR (HEAD (AGR 3SGPAST))
           (COMP (VP (SPEC NIL)
                (VBAR (HEAD OPEN))))))))))
```

The grammatical possibilities are: (OPENED IN)

The parser has gone as far as it can with this sentence.
You can choose to proceed in two ways:

Method 1 allows you to enter the next word to be completed.
The computer will then advance the parse as far as it can with this new word and then bring you back to this point.

Method 2 allows you to enter a list of words and the computer checks these words and tells you which of them could be used to advance the parse a further step. At that point you are returned to this point and you can choose to take one of those words or to check another list.

The sentence parsed so far is:
(THE GOLD KEY ON THE TABLE)

Now if you would like to use Method 1 type 1.
If you would like to use Method 2 type 2.
If you would like to quit this program type 0.
Enter your selection now:
1

(ENTER THE NEXT WORD) (opened the door)

This word was not in the dictionary
DOOR

Do you wish to enter it? Choose (y) or (n)
(y)

It will be entered now.

Enter the categories for this word.
Possibilities: DET PRO PREP ADV N ADJ V CONJ

Enter the category in list form (i.e., enter '(N)'). If a word belongs
to more than one category make this a list also (i.e., '(DET PRO)')

(N)

Enter the features for the word that have values.
Your choices are:
FOR N: NUMBER

Remember that all values for different interpretations of the
word must be entered in the same dictionary entry.
Enter these features with their appropriate arguments in list form.
For example '(TAKES (SGCT) NUMBER (SG))'.

(NUMBER (SG))

Enter the toggled features for the word.
Possibilities are:
FOR N: CARDINAL, MASS, COUNT, PROPER, PRE-DET, MENSURAL

Remember to include all that apply to this word in all of its meanings.
Enter them in list form like '(ART CENTRAL-DET)'

(COUNT)

Enter the root of the current word.
Only VERBS require roots, but other inflected words may have them.
If this word has not root, enter 'nil'.

nil

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
```

```
        (COMP
         (IP
          (SPEC
           (DP (SPEC NIL)
             (DBAR (HEAD THE)
                (COMP
                 (NP (SPEC NIL)
                   (NBAR
                    (ATTRIB
                     (DEGP (SPEC NIL)
                        (DEGBAR (HEAD NIL)
                            (COMP
                             (AP (SPEC NIL)
                               (ABAR (HEAD GOLD)))))))
                    (HEAD KEY)
                    (NU (SG))
                    (COMP
                     (PP (SPEC NIL)
                        (PBAR (HEAD ON)
                          (COMP
                           (DP (SPEC NIL)
                             (DBAR (HEAD THE)
                                (COMP
                                 (NP (SPEC NIL)
                                   (NBAR
                                    (HEAD TABLE)
                                    (NU (SG)))))))))))))))))
          (IBAR (HEAD (AGR 3SGPAST))
             (COMP
              (VP (SPEC NIL)
                (VBAR (HEAD OPEN)
                   (COMP
                    (OBJ
                     (DP (SPEC NIL)
                       (DBAR (HEAD THE)
                          (COMP
                           (NP (SPEC NIL)
                             (NBAR (HEAD DOOR)
                                (NU (SG))))))))))))))))))

  (DCL
   (CP (SPEC NIL)
     (CBAR (HEAD NIL)
        (COMP
         (IP
          (SPEC
           (DP (SPEC NIL)
             (DBAR (HEAD THE)
                (COMP
                 (NP (SPEC NIL)
                   (NBAR
```

```
                    (ATTRIB
                     (DEGP (SPEC NIL)
                        (DEGBAR (HEAD NIL)
                            (COMP
                             (AP (SPEC NIL)
                               (ABAR (HEAD GOLD)))))))
                    (HEAD KEY)
                    (NU (SG))
                    (ADJUNCT
                     (PP (SPEC NIL)
                        (PBAR (HEAD ON)
                          (COMP
                           (DP (SPEC NIL)
                             (DBAR (HEAD THE)
                                (COMP
                                 (NP (SPEC NIL)
                                   (NBAR
                                    (HEAD TABLE)
                                    (NU (SG)))))))))))))))))
          (IBAR (HEAD (AGR 3SGPAST))
             (COMP
              (VP (SPEC NIL)
                (VBAR (HEAD OPEN)
                   (COMP
                    (OBJ
                     (DP (SPEC NIL)
                       (DBAR (HEAD THE)
                          (COMP
                          (NP (SPEC NIL)
                             (NBAR (HEAD DOOR)
                                (NU (SG))))))))))))))))))
```

The parser has gone as far as it can with this sentence.
You can choose to proceed in two ways:

Method 1 allows you to enter the next word to be completed.
The computer will then advance the parse as far as it can with this new word and then bring you back to this point.

Method 2 allows you to enter a list of words and the computer checks these words and tells you which of them could be used to advance the parse a further step. At that point you are returned to this point and you can choose to take one of those words or to check another list.

The sentence parsed so far is:
(THE GOLD KEY ON THE TABLE OPENED THE DOOR)

Now if you would like to use Method 1 type 1.
If you would like to use Method 2 type 2.
If you would like to quit this program type 0.

Enter your selection now:
1

Enter the next word: (easily)

```
(DCL
 (CP (SPEC NIL)
    (CBAR (HEAD NIL)
        (COMP
         (IP
          (SPEC
           (DP (SPEC NIL)
              (DBAR (HEAD THE)
                  (COMP
                   (NP (SPEC NIL)
                      (NBAR
                       (ATTRIB
                        (DEGP (SPEC NIL)
                            (DEGBAR (HEAD NIL)
                                 (COMP
                                  (AP (SPEC NIL)
                                     (ABAR (HEAD GOLD))))))
                       (HEAD KEY)
                       (NU (SG))
                       (ADJUNCT
                        (PP (SPEC NIL)
                            (PBAR (HEAD ON)
                               (COMP
                                (DP (SPEC NIL)
                                   (DBAR (HEAD THE)
                                       (COMP
                                        (NP (SPEC NIL)
                                           (NBAR
                                            (HEAD TABLE)
                                            (NU (SG)))))))))))))))))
          (IBAR (HEAD (AGR 3SGPAST))
              (COMP
               (VP (SPEC NIL)
                  (VBAR (HEAD OPEN)
                      (COMP
                       (OBJ
                        (DP (SPEC NIL)
                           (DBAR (HEAD THE)
                               (COMP
                                (NP (SPEC NIL)
                                   (NBAR (HEAD DOOR)
                                       (NU (SG)))))))))
                      (ADJUNCT
                       (ADVP (SPEC NIL)
                           (ADVBAR (HEAD EASILY)
                                 (COMP NIL)))))))))))))
```

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP
        (SPEC
         (DP (SPEC NIL)
           (DBAR (HEAD THE)
              (COMP
               (NP (SPEC NIL)
                 (NBAR
                  (ATTRIB
                   (DEGP (SPEC NIL)
                      (DEGBAR (HEAD NIL)
                          (COMP
                           (AP (SPEC NIL)
                             (ABAR (HEAD GOLD)))))))
                  (HEAD KEY)
                  (NU (SG))
                  (COMP
                   (PP (SPEC NIL)
                      (PBAR (HEAD ON)
                        (COMP
                         (DP (SPEC NIL)
                           (DBAR (HEAD THE)
                              (COMP
                               (NP (SPEC NIL)
                                 (NBAR
                                  (HEAD TABLE)
                                  (NU (SG)))))))))))))))))
        (IBAR (HEAD (AGR 3SGPAST))
           (COMP
            (VP (SPEC NIL)
              (VBAR (HEAD OPEN)
                 (COMP
                  (OBJ
                   (DP (SPEC NIL)
                     (DBAR (HEAD THE)
                        (COMP
                         (NP (SPEC NIL)
                           (NBAR (HEAD DOOR)
                              (NU (SG)))))))))
                 (ADJUNCT
                  (ADVP (SPEC NIL)
                     (ADVBAR (HEAD EASILY)
                        (COMP NIL)))))))))))))
```

The parser has gone as far as it can with this sentence.
You can choose to proceed in two ways:

Method 1 allows you to enter the next word to be completed.
The computer will then advance the parse as far as it can with this new word and then bring you back to this point.

Method 2 allows you to enter a list of words and the computer checks these words and tells you which of them could be used to advance the parse a further step. At that point you are returned to this point and you can choose to take one of those words or to check another list.

The sentence parsed so far is:
(THE GOLD KEY ON THE TABLE OPENED THE DOOR EASILY)

Now if you would like to use Method 1 type 1.
If you would like to use Method 2 type 2.
If you would like to quit this program type 0.
Enter your selection now:
0

NIL

>

Chapter 3

THE GRAMMAR

The linguistic theory I have adopted to guide construction of my grammar is discussed in this chapter. I explain my implementation of particular concepts from this theory and the implications of this approach for building a computational grammar of English. Some example parses are given to demonstrate the range of structures the grammar can produce.

3.1 Motivation

This project aims to provide disabled users with a communication tool they can use for every day speech. Because of this it is necessary that the system be able to handle a wide variety of sentence structures; the grammar must be complete. Up to now, the biggest objection to using grammars for augmentative communication is that a sufficiently complete one is thought to be difficult to construct by hand. I have confronted this objection by making my grammar the embodiment of a syntactic theory from which I can derive an abstract, generalized description for a multitude of structures. This model of syntax is called X' pronounced "X-bar") theory and is borrowed from Government and Binding Theory. Its conventions make a complete grammar easy to construct and modify, while also providing a mechanism to describe the specific restrictions on what kinds of constituents can occur where. These restrictions are crucial to this project because the success of the syntactic predictor depends on it being able to eliminate categories that are not possible in a particular context. Thus, the use of X' syntax facilitates both completeness and restrictiveness in the grammar for this system.

3.2 X' syntax

All of the three most popular syntactic theories, Government and Binding (GB), Generalized Phrase Structure Grammar (GPSG), and Lexical-functional Grammar (LFG), have adopted forms of X' theory because of its explanatory power (Sells, 1985). This power comes from abstracting away the content of the phrase structure rules I discussed in Chapter 2 so that only the structural description is left.

Consider the phrase structure (PS) rules below:

(1) NP --> N

NP --> N PP

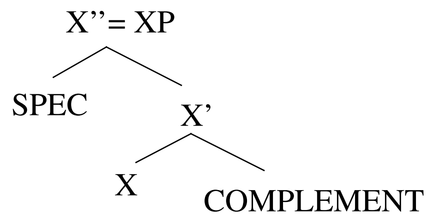
VP --> V

VP --> V NP

VP --> V PP

Notice that these rules serve two purposes: to tell what particular constituents the phrases on the left hand side of the rule can be broken into and also to give the position, or structure, of these constituents. Notice that the structure is similar among different phrases, for example both an NP and VP can be rewritten as just an N or V, respectively. In addition, they can both be rewritten with the N or V plus another constituent to the right. Thus, there is a fair amount of uniformity in the structures of different kinds of phrases. X' Theory tries to capture this similarity by claiming that the basic syntactic structure is given by the following template:

(2)



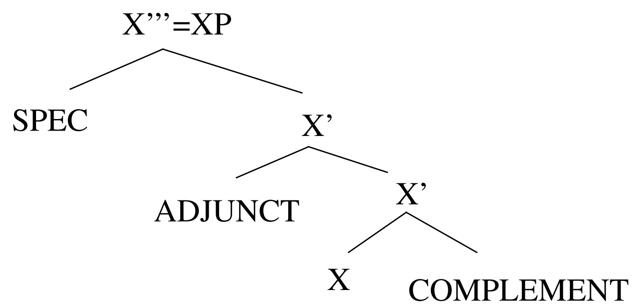
This generalized structure is motivated by the similar patterns found in the internal structure of different kinds of phrases (i.e., noun phrases, prepositional phrases, verb phrases): they all have a head constituent, complements and various other modifiers that can come either before or after the head. In the template, the head is represented by the variable X. This is the element that gives the phrase its character; for example, the head of the NP is an N, the head of a PP is a P, and the head of the VP is the V. The entire phrase is said to be a "projection" of the head; a structure built up using this template is called a "maximal projection" because the entire template structure has been used. It is also referred to as an "X-double-bar", reflecting the fact that it is the highest level of the template and includes all head modifiers.[1]

I have adopted a formulation of X' theory that takes the intermediate X' level as the site where modifiers like adjectives and prepositional phrases are attached (Radford, 1988). The modifier, also called an adjunct, can be attached on either side of the X' so that it could be in either pre-head, as is the case with adjectives, or a post-head position as with

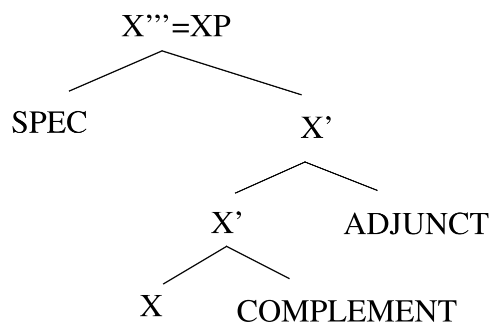
1. In this formulation, I am assuming that a maximal projection can only have one double-bar level.

prepositional phrases. The X' is recursive in that modifiers expand it into another X' level, making the structure[2]:

(3)



OR



This means the intermediate X' level plays a crucial role in the syntactic structure. But this level is also the most non-intuitive: we would normally talk about a noun as the head of a phrase and the entire noun phrase, but we would not consider any predefined intermediate level. There is convincing evidence for the existence of this level, especially with coordination data.

Coordination is when two words or phrases are conjoined with a conjunction like "and" or "but". Only constituents, which are structural units whose lexical items are

---

2. In structure (2) "Specifier" and "Complement" are not syntactic categories or constituents. They are only labels describing the function of the constituents that hold these positions in the structure.

immediately dominated by the same node in the syntax tree, can be conjoined, and furthermore, the constituents must be of the same type (i.e., they both must be NP's, VP's, N's, etc.). Consider the data below:

(4) *John rang up his mother and up his sister.

[Constituent and non-constituent conjoined]

(5) John rang up his mother and his sister.

[Two constituents conjoined]

(6) *John wrote a letter and to Fred.

*John wrote to Fred and a letter.

[Constituents of different typed conjoined]

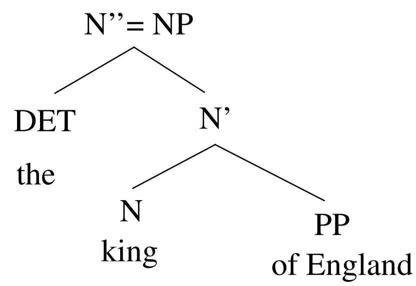
(7) John wrote to Mary and to Fred.

John wrote a letter and a postcard.

[Constituents of the same type conjoined]

Now consider the noun phrase "The king of England" in the following example, taken from (Radford, 1988, p. 174-175). Under X' theory, this noun phrase has the following structure:

(8)



The question is how we know that [king of England] is a predefined constituent called N'. In response to this question, observe the following coordination data:

(9) Who would have dared defy the [king of England] and [ruler of the Empire]?

(10) Who would have dared defy the [leader of the army] and [king of England]?

Given the restrictions on what kinds of structures can be conjoined, the grammaticality of (9-10) show that [king of England] must be a constituent to itself. Notice too that sentences (11) and (12) would not be ungrammatical. In these cases, the conjoined constituent is a N'' (or NP) rather than a N'.

(11) Who would have dared defy [the king of England] and [the ruler of the Empire]?

(12) Who would have dared defy [the leader of the army] and [the king of England]?

It is also possible to find evidence for the intermediate N' constituent in Shared Coordination data. This is a type of coordination wherein a constituent is "shared" between two conjuncts. For example in the sentences:

(13) John walked (and Mary ran) [up the hill.]

(14) John will, and Mary may, [go to the party.]

Analysis of these kinds of data have shown that Shared Coordination is only possible where the shared string is a possible constituent of each of the conjuncts (Radford, 1988, p. 78). Because of data like (15-16) below, we can conclude that [king of England] must make up a constituent in its own right, which is separate from either the head noun or the entire noun phrase.[3]

(15) He was the last (and some people say the best) [king of England].

(16) He was the last (and some people say best) [king of England].

Thus, it seems that there is good reason to accept the intermediate level posited in structure (2).[4] The most basic construction of this intermediate level (i.e., not considering adjuncts) includes the phrasal head and its complement, which is also called its argument and which is structurally its sister. The head of the phrase subcategorizes for its complement, meaning that it requires a particular kind of complement to occur with it. For example if the phrasal head were a verb, it would subcategorize for a particular kind of object (often a noun phrase), and if the head were a preposition, it would subcategorize for a noun phrase. I will discuss more about the kinds of complements heads can subcategorize for below and in Chapter 4.

---

3. Note that in (15-16), it is not possible to call [king of England] an NP in itself because then there would be no way to explain the structure of "the last" in these sentences. The essence of shared coordination is that the shared string completes the constituent of the thing that shares it. Hence, the N' [king of England] completes the NP [the last king of England].

4. Notice that this is not evidence for a unique N' constituent. It does prove that there is some kind of intermediate level between the head X and the entire phrase, or maximal projection. Since this intermediate level is not unique, it is possible to adjoin various adjuncts to it, as is seen in (3). The X' levels are simply a way of describing a structure that is more than just the head, but does not include all of the constituents associated with the head.

The sister to X' is called a specifier, which has the function of expanding the X' completely into a maximal projection. There is some discussion among X' theorists about what kind of constituent can occur as a specifier. The position I am accepting for constructing this grammar is elucidated by Thomas Ernst, who treats the position as "the response of syntax to the need to give special status to some particular peripheral element: demonstratives, subjects, etc. (Ernst, 1990, p. 25)."[5] He proposes that the specifier position is to be used to ensure that certain elements are always phrase-initial position; for example, following data borrowed from (Ernst, 1990, p. 9) shows that it is not possible to reorder any words to produce emphasis:

(17) a. A fancy new car.

b. A new fancy car.

(18) a. The many honest men.

b. *The honest many men.

For his definition then, the specifier position is one that is used to describe ordering constraints like these. As such, it does not involve restrictions on the kinds of structures can occur in this position (i.e., specifiers do not have to be maximal projections, contrary to a widely known Chomskian theory).[6] This interpretation is also described in (Quirk et. al., 1985), where it is explained that some words, such as determiners and particles, are

---

5. Presently I will show that the subject occupies the specifier position when the X' template is applied at the sentence level.
6. Chomsky's Maximality principal described in (Chomsky, 1986) says that all non-head elements in a maximal projection (i.e., instance of the X' template) must themselves be maximal projections. This view is accepted by Susan Rothstein and other syntactitians working in Chomsky's tradition.

"single tokens, complete to themselves." I have adopted this interpretation of the specifier position because often the words that show up in specifier position are those that Quirk calls "single tokens" and therefore they do not form maximal projections of their own. This is not always the case, for I will show that a sentences's subject occupies a specifier position; however, the single tokens occur often enough for it to be computationally inefficient to require that all specifier positions be maximal projections. In my grammar, the specifier position can contain either a lexical item or a maximal projection, depending on the head of a particular specifier's projection.

In the syntactic "template" I have been discussing (i.e., structure (2) above) only the head element is required. The specifier, complement, and adjuncts are all optionally present for any given instantiation of the template. The presence of the complement is determined by the head itself.[7] For example consider the case of the head being a verb, which would make the XP a VP. A transitive verb like "kill" requires a complement, as shown in the following data:

(19) John killed Mary.

*John killed.

---

7. It is argued in (Ernst, 1990) that the head can also determine what is in the specifier position, meaning that it is able to do what I will call "backwards subcategorization." This suggestion falls out from the fact that Ernst is not accepting the DP analysis of the noun phrase (discussed below) but still needs to account for the agreement facts that raised the argument for the DET functional head in the first place. As will become clear in what follows, I have accepted the DP analysis of the noun phrase and consequently, I explain the agreement as the DET subcategorizing (forward) for the noun phrase. This is necessary to do prediction, because it would not be useful to have to wait for the head to be entered before being able to determine if what has already been parsed in the specifier position is properly licensed. Instead, I specify particular entities that can exist in the specifier position based on knowing the instantiation of XP. For example, if the XP is a NP I know that only certain elements can occur as prenominal modifiers and these are coded into the grammar

Conversely, an intransitive verb like “cry” does not take a complement:

(20) John cries.

*John cries Mary.[8]

The power of the phrasal head to dictate its complement is called subcategorization.[9] This ability is common to all phrasal heads, although there are some behaviors phrasal heads may or may not exhibit that have prompted syntacticians to distinguish different types. Here I will adopt the taxonomy explicated by Susan Rothstein (Rothstein, 1991), who distinguishes three kinds of heads: lexical, functional, and minor.

Lexical heads are those like verbs or prepositions. They determine the character of their maximal projection so that if the head is a V, the projection is a verb phrase (VP), or if the head is a P, the projection is a prepositional phrase (PP). These words have very specific requirements on the number of complements they must have and this number must always be satisfied for the phrase to be realized.[10]

The second kind of head is called a functional head because of its functional role in a phrase. These heads determine the nature of their maximal projection just as the lexical head does, but they are not necessarily realized as a lexical item. For instance the INFL, which is considered the head of a sentence, is typically said to be realized by the tense and

8. In the sentence “John cries for Mary” the “for Mary” would be an adjunct preposition, rather than a complement.
9. The phrasal head selects its complement, but it does not select for adjuncts. They are present only as “extra” information in the sentence.
10. Here I am alluding to the fact that they have theta-grids and that all theta roles must be discharged. This comes from GB’s conception that syntax is a projection of lexical properties and so each head gets exactly the number of arguments that is specified for it in the lexicon. The “Theta Criterion” of GB ensures that heads and their arguments are in proper distribution.

agreement of the main verb, and sometimes as a modal. Since the head of the sentence is called INFL, X' theory describes the canonical sentence as an IP, or "Inflection Phrase."[11] The 4 major heads in this category are INFL, which holds inflection, DET, which holds the determiner and also determines agreement, and COMP, which holds the complementizer "that" in embedded clauses and whose specifier position holds WH-question words. I will discuss each of these in more detail in the implementation section below.

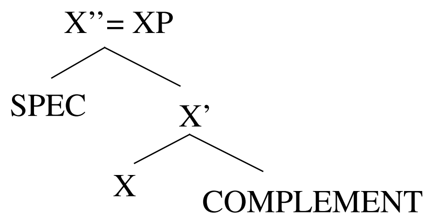
The minor heads are also functional heads in the sense that they are not frequently lexicalized, but unlike the previous two head types, the minor heads do not determine the nature of their projection. This is done instead by the complement they subcategorize for depending in some cases on their position in the sentence. Typical minor heads are degree words like "too" or "as" and therefore the head is called DEG, for degree. The "degree phrase" is discussed more completely in the implementation section below.[12]

The grammar I have built is based on these heads and the structures they generate via subcategorization. Since there is a finite number of instantiations for these head types, I have eliminated the need for Phrase Structure rules that tell exactly what constituents go where. Instead the structure of (2), reproduced below, provides the order and relationships of constituents to one another and all other information comes from the requirements (i.e the subcategorization) of the specific words themselves.

---

11. By "canonical" I am referring to a top-level sentence, rather than an embedded one.
12. Conjunctions are also said to be minor functional heads. Although they are not implemented here, they could be done in a way similar to the degree phrase.

(2)

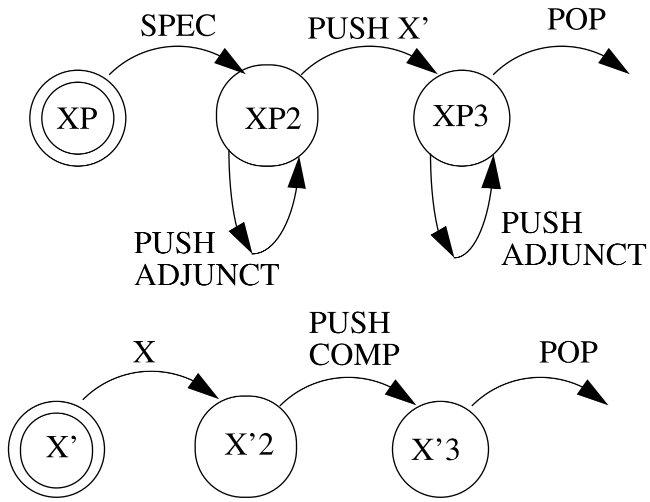


The grammar has both top-down and bottom-up motivation in the sense that while the template structure must be satisfied, the input itself determines how that will be done through subcategorization. This simplifies writing the grammar because now its not a question of including enough PS rules, but of providing the structure and allowing the subcategorization to determine when it is appropriate.

3.3 Implementation of X' Theory

Recall in my discussion of the ATN formalism that I characterized a PUSH arc as the mechanism's way of handling frequently occurring constituents such as noun and prepositional phrases. The PUSH arc gives the ATN more power because it allows recursive processing (i.e., a given network can refer back to itself, as in the PS rule NP --> NP PP). It was created for its computational power; without recursion, grammars are enormous and frequently redundant. In this project I propose a linguistic motivation for the use of PUSH arcs- namely as the means for creating a projection in the X' theory template. All maximal projections are the result of 2 PUSH arcs: a PUSH arc for the X' level and a PUSH for the X'' level. Each PUSH arc adds more structure to the level below, so that the implementation of the template in (2) is done with the following networks:

(21)



For every instantiation of X there is a pair of networks such as these. Each time a maximal projection appears in a network, it is done by PUSHing for that XP. Within that XP, an X’ constituent is PUSHed for because it is a projection of the head. Only the head is a non-PUSHed element, just as only the head is a non-maximal projection in a rule.[13]

In Figures 3.1 and 3.2 on the following pages, it is clear how the structural template of X’ theory, as implemented in the two networks in (21) has greatly improved the task of writing a grammar. Figure 3.1 shows the series of networks that make up the entire grammar I have implemented. Figure 3.2, reproduced from (Bates, 1983, p. 217-219), shows the networks for the LUNAR grammar, which was one of the earliest and most complete ATN grammars. The X’ theory grammar has a greater coverage than the

13. This explanation recalls Chomsky’s Maximality constraint; however as I mentioned previously, I am not adopting this because of my position that specifiers can also be lexical items. Hence, the statement about heads being the only non-PUSHed element must be qualified in the case of a specifier. Sometimes specifiers are maximal projections, and therefore PUSHed constituents, and sometimes they are unprojected words. Not requiring specifiers to be maximal projections is computationally preferable because when the specifier is a single word unnecessary PUSHes do not need to be done only for the sake of the Maximality Constraint.

Figure 3.1 X’ Theory grammar

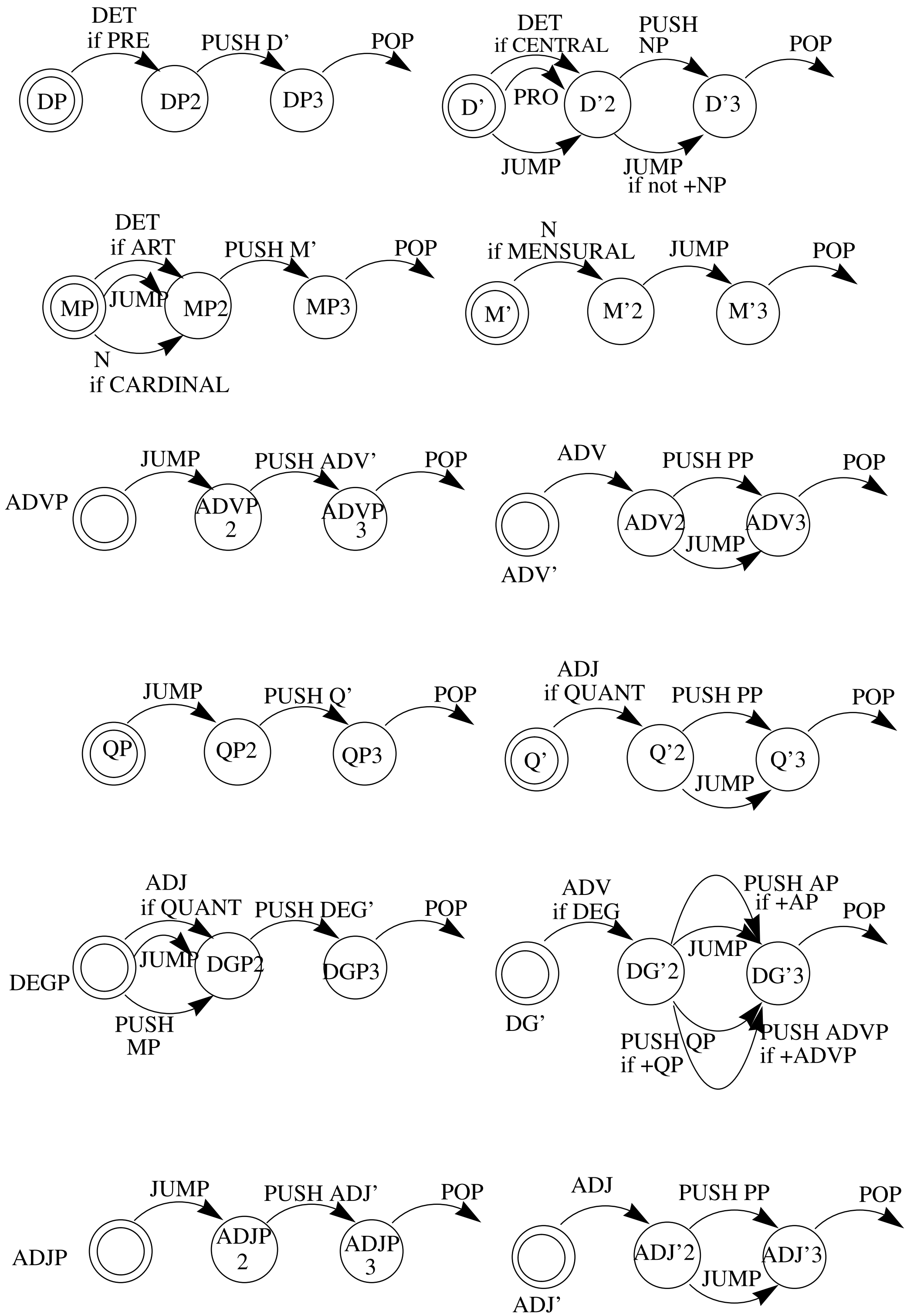


Figure 3.1 X' Theory grammar (continued)

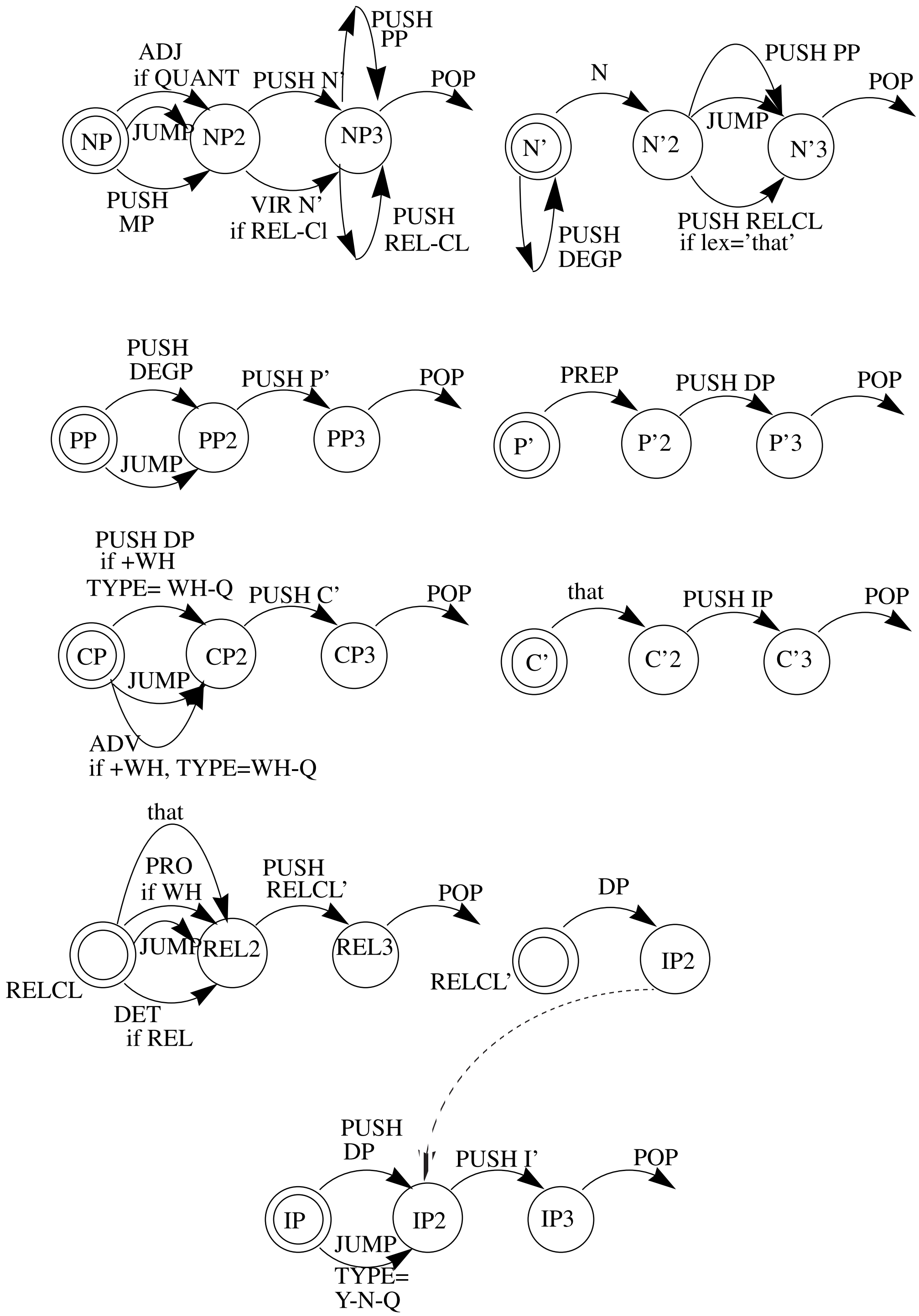


Figure 3.1 X' Theory grammar (continued)

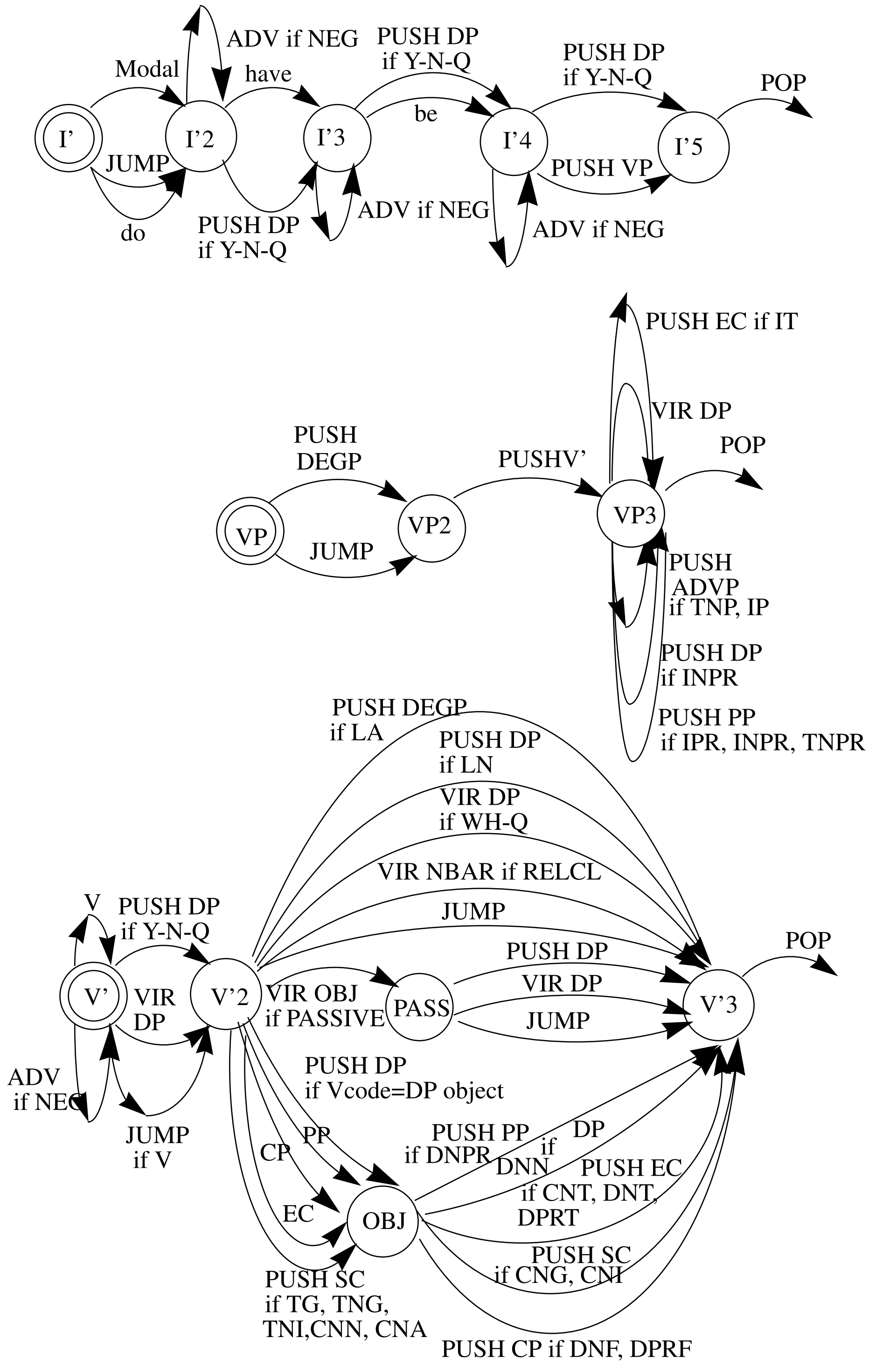
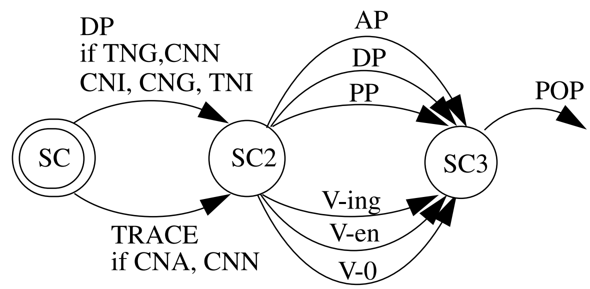


Figure 3.1 X' Theory grammar (continued)



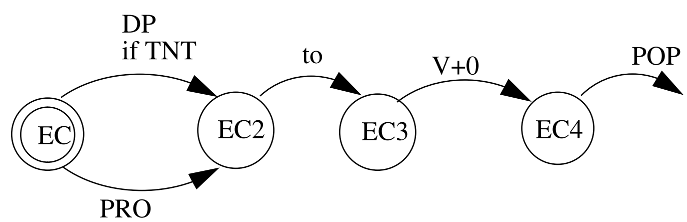


Figure 3.2 LUNAR grammar

Figure 3.2 LUNAR grammar (continued)

Figure 3.2 LUNAR grammar (continued)

LUNAR grammar and is a more sleek implementation. This makes modifying it to cover more syntactic phenomenon (e.g., topicalization, ellipsis, etc.) easier because it is immediately clear where new pieces of grammar need to be added to account for these structures.

In the following sections I discuss the details of implementing network pairs for each of the heads. In Chapter 5, I will further discuss the implications and deviations of this implementation from Government and Binding Syntax, whence I borrowed the X' theory grammar framework.

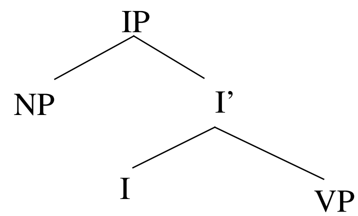
3.3.1 Sentence level

As I mentioned previously, GB theory posits a functional category called "INFL" for "inflection" in order to apply the X' template to the sentence level. This category holds the tense and person-number agreement information for the sentence. This information can be lexicalized in the form of a modal if the clause is finite or as a "to" in the case of non-finite (e.g., infinitival or participial) clauses.[14] INFL can not hold a modal and a "to" at the same time because sentences must be either finite or non-finite. It is possible for INFL to be unlexicalized, as when there is just a single verb in the sentence. In this case the agreement and tense are considered to be in INFL, but lexicalized only on the main verb.

The INFL head and the X' template of (2) produce an analysis for a sentence traditionally analyzed as [NP VP] with the following form:

14. The words "have" and "be", which were categorized with modals in transformational grammar, do not occur in INFL. They are considered to be verb-inflections rather than INFL-inflections and as such, they appear as part of a complex VP.

(22)



The head of this structure is INFL, the complement of the head is the VP and the specifier of I’ is the NP. The overall structure is a maximal projection called an INFLection-phrase or IP.

A problem with this structure arises when sentences have embedded IP’s like

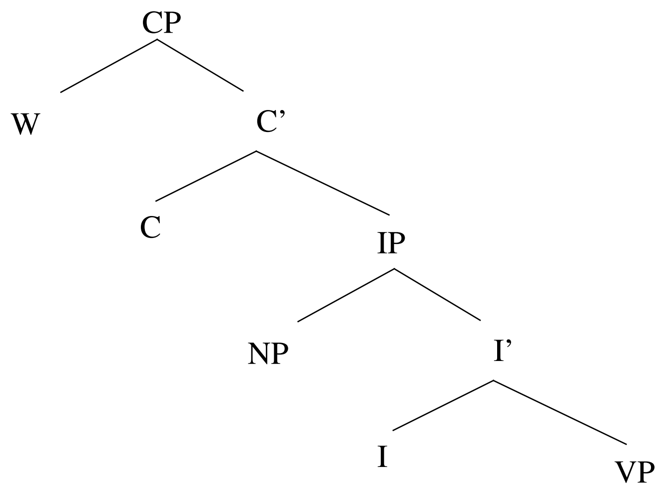
(23) The committee may insist [that the chairman resign].

because the IP structure in (22) can not accommodate the “that”, which is a complementizer that introduces the embedded clause. It would not be correctly analyzed as a determiner like “the” because it introduces the entire embedded sentence. Similarly, it would not be an object of “insist” because it has no reference as an independent pronoun in this sentence. This problem is solved in X’ theory with the functional head COMP, or Complementizer. The COMP is the head of the overall sentence, holding the “that” in embedded clauses or being empty for top-level sentences.[15] The IP is in the complement position in the template, making the structure of sentences, called CP’s for “Complementizer Phrase”, like the following:

---

15. All sentences are CP’s, but because top level sentences rarely have a lexicalized Complementizer, they are sometimes referred to simply as IP’s

(24)



Making the sentence head a Complement is supported by the fact that Complementizers influence the content of the INFL node. Any clause that has a Complementizer must have an INFL that is compatible with it in terms of being finite or non-finite. For example a non-finite complementizer like "for" cannot introduce a sentence with a finite INFL:

(25) *They are anxious [for you make up your mind.]

But a non-finite INFL is acceptable:

(26) They are anxious [for you to make up your mind.]

My implementation of sentences reflects the structure shown in (24). This implementation has one major deviation from the GB formulation of X' theory, this being the content of the INFL node itself. In the structure my grammar produces, INFL holds a code showing tense and agreement, as X' theory stipulates, but it also holds the auxiliary verbs "have" and "be". This is counter to the notion that each X can only hold one lexical element, which is a central argument for why "have" and "be" do not appear in INFL in normal X' theory (cf. (Radford, 1988, p. 312)).[16] Aside from this, the only difference between modals and these auxiliary verbs seems to be that they determine verbal inflection on the verb that follows them (i.e., "have" requires a participial verb, and "be" requires a gerundive verb). In an attempt to capture this, whenever these auxiliary verbs appear in my INFL, they are accompanied by the appropriate inflection marker. Thus, if a sentence includes all the auxiliary verbs (e.g., "The man could have been sleeping"), the INFL node my grammar will produce has the following form:

(27) (INFL (AGR 3SGPAST) (MODAL COULD) (HAVE EN) (BE ING))

The constituent AGR is the agreement marker wherein number and person agreement is combined with the verb tense. This is different from the standard X' conception of INFL as containing two binary variables: one for agreement and one for tense. I have combined these variables in order to account for lexical items whose agreement changes with their tense. Consider the sentences below:

(28) The man hit the ball yesterday.

[Intended past tense.]

(29) *The man hit the ball today.

[Intended present tense.]

---

16. GB does hold that "have" and "be" can appear in INFL at surface structure if there is no modal in INFL. This is the result of "have" or "be" raising into the INFL position from their original position as part of a complex VP. It is therefore not unheard of for these words to appear in INF; the major deviation in my implementation is that more than one of them can appear in INFL.

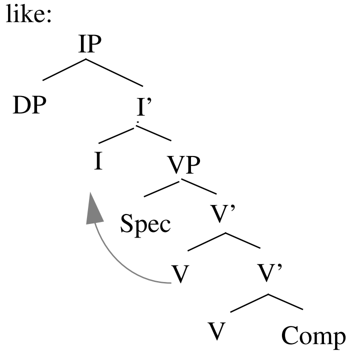
The problem here is that the same lexical item can be past or present, but have different agreement requirements. The sentence in (29) is ungrammatical because, in the present tense, “hit” can only agree with non-third person singular subjects. It was necessary to find a separate way to account for these two lexicalizations because it is not possible for the lexicon to have two separate entries for the same lexical item. By using these single variables, agreement could be preserved: it is possible to specify that “hit” can agree with 1SGPRESENT, 2SGPRESENT, 1PLPRESENT, 2PLPRESENT, 3PLPRESENT, and all of the PAST values (i.e., 1SGPAST, 2SGPAST, etc.).[17] The details of these lexicon entries are explained more thoroughly in Chapter 4.

Implementing the INFL in this way means that the Verb phrase only contains a single V which is the main verb in the sentence. This is in contrast to the branching complex VP structure that would be necessary to hold “have” and “be” when they occur.[18] Eliminating these branching structures means extra PUSHes have been eliminated and this

---

17. Prima facie it seems that an indication that “hit” can occur with all forms except 3SGPRESENT might be a better way to explain its distribution. The lexicon is set up so that the actual lexical entry allows for short cuts like this; however it is useful to be able to specify which exact combinations a word can occur with- especially in the case of the personal pronouns. This implementation seems to facilitate handling all agreement, even though some realism is lost through positing these “agreement codes”. Quid pro quo.

18. This structure would look like:



where the first branching verb, “have” in this case, would move into INFL position from its position shown here. (Zagona, 1988) makes the processing more computationally efficient. In preserving the participial agreement requirements of “have” and “be,” and also checking for these requirements in tests on the grammar rules, I have maintained the ability of these verbs to determine structure without any extra computation. The same result is produced; the branching structure is just different from what X’ theory predicts.

Subsequently I will refer to sentences as CP’s headed by COMP and sentences without complementizers as IP’s headed by INFL (cf. footnote 15).

3.3.2 Determiner Phrases

The noun phrase can be organized around a functional head of the same type as INFL. This was demonstrated by Steven Abney when he argued for a DET head that is lexicalized by the determiner (Abney, 1987). The rationale behind this analysis stems from the fact that the determiner can subcategorize for its complement like lexical categories and therefore must have a more central role in the noun phrase than simply specifier, as shown in (8). For example in English there are some determiners that either can or can not take complements. Consider:

(30) That is terrific.

[Complement not required.]

(31) *The is terrific.

[Complement required.]

(32) The boy is terrific.

[Complement required.]

(33) *A boys is terrific.

[Particular type of complement required.]

Thus, it is evident that the determiner acts just like the verb in shaping its projection.

The functionality of DET (i.e., its similarity to INFL) can be seen more clearly where the determiner not only specifies a complement, but also determines its form. For example in English, as was shown in (32-33), a determiner like "a" requires a singular noun to follow it, while a determiner like "the" will allow either a singular or plural noun. This is seen more clearly in languages which mark for agreement between the determiner and the noun. For example, in Turkish we see the following data, borrowed from (Abney, 1987, p. 49):

(34) a. el

"the/a hand"

b. (sen-in) el-in

"you-GEN hand-2s"

"your hand"

c. (on-un) el-i

"his hand"

In this example it must be noted that Abney considers the personal pronouns as lexicalizations of DET in that they function to determine the form of the noun.[19] In the

---

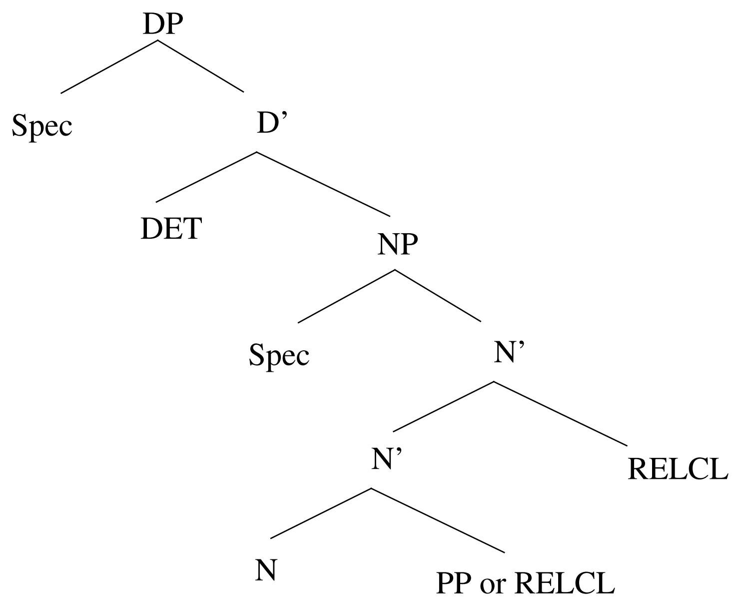
19. I will further discuss the idea of pronouns as determiners in what follows.

(34a), the generic, uninflected form of the word "hand," "el," is shown to have the indeterminate meaning "a/the hand". Datum (34b) shows that with the second person pronoun "your", the word for "hand" takes on a 2nd person inflection to agree with the pronoun: hence the form "el-in". A similar agreement is shown in (34c) where the third person pronoun-determiner requires a third person inflection on the head noun "hand". Thus, there is a relationship similar to that described by the INFL functional head between what serves as the determiner in DET and the noun. This example clearly shows that the noun is declined according to the specification of the determiner, just as INFL determines the inflection on the main verb at sentence level.[20]

An analysis of DET as the functional head of the noun phrase serves to unify the X' theory analysis of the noun phrase with that of the sentence. The structure of the DP, reminiscent of IP, is shown below. Note that a relative clause can serve as either a complement or an adjunct. This is discussed in more detail in section 3.3.5.2 of this chapter.:

20. Footnote 7 provides an alternative analysis of these data; however, for reasons put forth there, it is not the analysis accepted for this implementation.

(35)



The most useful effect of this structure for the purposes of this project is that with the specifier positions of DP and NP, there is extra structure to account for the various types of words that can occur in these positions. Without this structure they would have to be considered adjuncts before the determiner or between the determiner and noun; however, there would be no significant ordering among these words. Consider the following noun phrases

(36) a. [DET a] [NP-SPEC dozen] roses

b. [DP-SPEC all] [DET 0] [NP-SPEC six] men

c. *[DP-SPEC six] [DET 0] [NP-SPEC many] men

d. [DP-SPEC all] [DET the] [NP-SPEC six thousand] men

e. [DP-SPEC all] [DET the] [NP-SPEC many] men

d. [DP-SPEC many] [DET the] [NP-SPEC 0] men

Furthermore, if adjectives can come between determiners and nouns in relatively free order, why can't other words such as the quantifier "many"? Recall the example given in (17-18), reproduced here:

(17) a. [DET A] [ADJS fancy new] car

b. [DET A] [ADJS new fancy] car

c. [DET The] [Q many] [ADJ honest] men

d. *[DET The] [ADJ honest] [Q many] men

It seems therefore, that there are certain types of words that must appear in particular positions within the noun phrase, and the extra specifier positions provided by (35) allow provide for these words. Following this analysis, I have implemented a structure wherein the Determiner is the head of the noun phrase. This means the noun phrase is a projection of Det and hence called a DP (i.e., determiner phrase). I will refer to what have traditionally been called noun phrases as Determiner Phrases throughout the rest of this work.

In addition to arguing that Det is the head of the noun phrase, as I mentioned previously, Abney argues for generating pronouns in the Determiner position. While this is not to say that pronouns are determiners, it is a way of accounting for the fact that like determiners, pronouns have a primarily functional status. They provide agreement features like number, person, and gender and therefore influence the form of the verb that follows them. As we have seen in Turkish, they can also influence the form of the noun that follows them when the language is a highly inflected one.

Pronouns are implemented in this grammar in the same position as determiners. This has the nice effect of allowing pronouns to be used with non-empty noun heads, as in:

(37) We students are tired.

Here, just as in Turkish, the number of the noun must agree with that of the pronoun (cf. "*We student are tired." and "*We student is tired.").

This implementation also allows the possibility for determiners that do not require NP complements, (i.e., "that") to stand alone in DP's:

(38) That is ridiculous.

Note that the specifier position of DP's contains only certain kinds of determiners like "all" which can precede the articles. The other positions in the X' theory template for DP's are filled as follows: articles and other pronouns are the only elements in the head of DP position. There is only one kind of complement for DP's: a NP. It is possible to have adjective phrases occurring before the NP (cf. next section) but these occur in adjunct position rather than complement position.[21]

3.3.3 Degree Phrases

To accompany his interpretation of the DP as a projection of a functional head, Abney posits an abstract head to describe adjective phrases and adverb phrases. Abney's explanation of this head, which he calls DEG, for "degree", makes it a head of the kind

21. This is a significant point of departure from Abney's discussion of the structure of the DP. His makes the adjective a complement to DP, a move motivated by his opinion that a structure where an X' expands into an X' is undesirable. I have adopted this very structure based on the arguments of (Radford, 1987, p. 179-196). Consequently, I am inclined towards positing that the adjective phrase is in adjunct position, attached to an X'.

that Rothstein calls functional. Rothstein herself further refined the analysis of the DEG head by calling it a "minor" functional head. While empty with simple adjectives and adverbs (i.e., phrases such as "white hair" or "run quickly"), it is lexicalized by words like "how", "this", "that", "so", "too", "as", "more", "less", "all", "most", and "least".[22] Like INFL and DET, DEG can also give inflection to its head because it is also the place where comparative -er and superlative -est are specified.

With his functional head, Abney tries to capture the generalities between adjective and adverb phrases, claiming that they are the projections of the same node. The structure he posits, has an adjective phrase being the only kind of complement and the adverb phrase as either a subcategory of adjective phrases or in the specifier position of an otherwise empty structure. Rothstein argues against this analysis in her characterization of the degree phrase as a "minor" functional category.

As a minor category the DEG head does not determine the nature of the phrase (i.e., whether it is an adjective or adverb phrase). Instead, it subcategorizes for a particular kind of complement based on its position in the sentence, and this specifies the nature of the overall maximal projection. Rothstein accepts two kinds of complements that a DEG could call for: adjective and adverb. Examples where the same degree word can occur with different complements according to their sentence position are given below:

(39) Adjective phrases as part of a noun phrase:

a. too red shirt

---

22. Abney explains that the head being empty is not problematic since the same thing happens to DET when there is no overt determiner in noun phrases.

b. more rich man

c. so sleepy kitten

(40) Adverb phrases as part of a verb phrase:

a. ran too quickly

b. more frequently seen

c. so completely exhausted

In addition to adjectives and adverb that occur with degree words, there is a class of words, including “many”, “much”, “few”, “several”, and “little”, which Abney and Jackendoff before him call quantifiers (Jackendoff, 1977). On the basis of this distinction Abney posits a third complement that the DEG could subcategorize for, making it a quantifier phrase (QP), such as:

(41) a. too many

b. as little

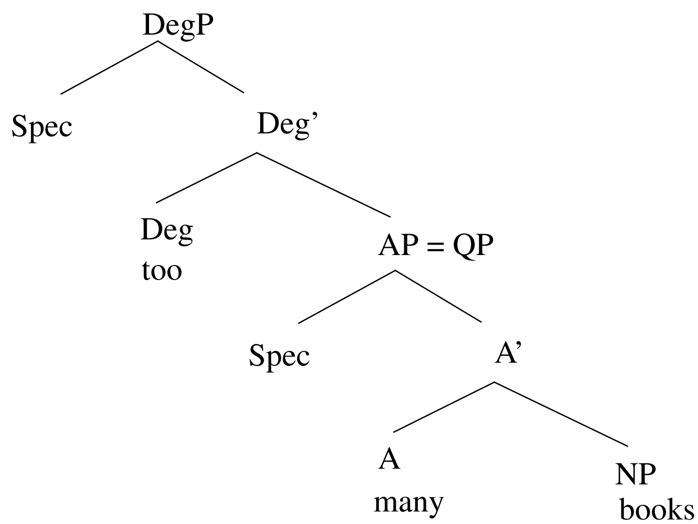
c. +er few --> fewer

The quantifier can occur in the complement position of the degree phrase in the same way as adjectives and adverbs do. Rothstein argues against this analysis by denying the distinction that the quantifier is a special kind of adjective. Consider the sentence below:

(42) Tom bought [too many books to carry them all home].

Rothstein holds that the “many” heads a QP which is in all other respects a noun phrase. In doing this she is envisioning a structure like the following, which is taken from (Rothstein, 1991):

(43)



By collapsing the distinction between quantifiers and adjectives, Rothstein is able to claim that the structure in (43) suggests that sentences like (44) below should be grammatical.

(44) *I met too stubborn children to help them.

She claims that because (44) is ungrammatical, “many” in (43) can not be analyzed as an adjective. In addition, she believes the data in (45) shows that “many” and “one” occur in the same structural position.

(45) a. many books

b. two books

c. one book

d. a book

e. the book

f. *book

g. *green book

She says that since “one” satisfies the requirement that singular nouns have a determiner, and therefore must be a determiner, that “many” must also be a determiner.

I argue that Rothstein is correct in her analysis that words like “many” are not like adjectives, but I am unconvinced that they are determiners and especially that it is unreasonable to consider them “special” adjectives. If they were “special” it is reasonable to predict that (42) is grammatical while (44) is not. Her argument based on data in (45) is particularly unconvincing because “many” requires a plural noun phrase and plural noun phrases do not need determiners. Therefore it is possible for the data in (46a-c) to be grammatical and for the words “many”, “few”, “several”, etc., to be something other than determiners:

(46) a. many books

b. few books

c. several books

d. *many book

e. *few book

f. *several book

Thus, I am accepting Abney's analysis of these kinds of words as quantifiers and implementing them as a special kind of adjective that can be a complement to the DEG phrase. Consequently, my DEG phrase has three possible complements: adjective phrases, adverb phrases, and quantifier phrases. Each of these phrases is a full maximal projection which can have prepositional phrases as complements.[23]

In allowing the DEG word to subcategorize for a particular complement, I am accepting Rothstein's analysis for my implementation of degree phrases. A major deviation of my implementation from her work comes from the difficulty in representing a structure whose character is not determined until the complement is parsed. Part of this problem stems from the fact that in suggesting this behavior for her minor heads, Rothstein must violate one of the most basic tenets of X' Theory: that a head always determines the category of its projection. She does this to maintain "forward" subcategorization between the head and the complement, rather than allow "backwards" subcategorization (cf. footnote 7).

For reasons discussed previously relating to the requirements of prediction, I cannot allow backwards subcategorization and therefore must agree with Rothstein's analysis that a degree head chooses its complement which in turn determines the character of the phrase. This does not necessitate a completely "forward" subcategorization because

23. The implementation of prepositional phrases as complements of ADJP, ADVP, or QP is motivated by the discussion in (Radford, 1987, p. 241-246). This discussion does not conceive of these phrases as part of DEGP and therefore represents the overall structure of the ADJP, ADVP, and QP differently from what I have describing. Nevertheless, I have found no other explanation for what can serve as constituents in an ADJP, ADVP, or QP and so I have adopted the portion of Radford's analysis that is appropriate. As the implementation stands, the specifier and adjunct positions of these phrases are always empty.

the complement that is chosen is often dependent on the structural position of the degree phrase. Since I know what that position is, it is possible to only allow particular complements to occur in particular places (e.g., ADVP's do not occur as noun modifiers). In this case, subcategorization is not completely dependent on the head of the DEG phrase and so it does not have the same functional importance as the DET and INFL functional heads. The appeal of the DEGP analysis is that it provides a nice way to capture the similarities between the adjective, adverb, and quantifier phrases.

Rothstein suggests that depending on the complement of the DEG phrase, it is called either an ADVP, ADJP or QP. This is a problem for the implementation because the complement is analyzed after PUSHing for a particular maximal projection, and therefore it is not possible to allow the character of the maximal projection to be determined after analyzing the complement. This means the structure of my degree phrase appears more like that of Abney's, in that it is always labeled "DEGP," rather than AP or ADVP as Rothstein would prefer. The character of the complement is explicit in the structure produced so that no explanatory power is lost.

As part of my implementation of DEGP's and my decision to categorize quantifiers as special kinds of adjectives, I allow them to occur in the specifier position of the DEGP. This accounts for data like the following:

(47) a. I have [SPEC-DEGP much] [DEG too] [COMP-DEGP much]
work to do.

b. A [SPEC-DEGP few] [DEG too] [COMP-DEGP many] men attended the dance.

c. A [SPEC-DEGP few] [DEG 0] [COMP-DEGP 0] men attended the dance.

d. [SPEC-DEGP Several] [DEG 0] [COMP-DEGP 0] men attended the dance.

These data could also be accounted for if the quantifier were in the specifier of NP position. This would also allow a simplified structure for (47c-d) because there would not be an empty headed degree phrase:

(48) a. A [SPEC-NP few men attended the dance.

b. [SPEC-NP several] men attended the dance.

Because of this, I have also allowed quantifiers to occur in the specifier of NP position. This eliminates unnecessary computation and solves Abney's problem of having an empty NP specifier (cf. (Abney, 1987, p. 341). I will discuss NP's in more detail in the following section.

One final type of phrase that Abney singles out is the "mensural phrase". These phrases have a cardinal or ordinal determiner and a mensural noun[24] as their head. Examples are:

(49) a. six weeks

b. ten times

c. a dozen

---

24. A noun which specifies a countable unit, such as "dozen", "bushel", "bundle", "feet".

These phrases are closely related to the head of the DEG phrase when it is lexicalized, as in

(50) a. ten times as quickly

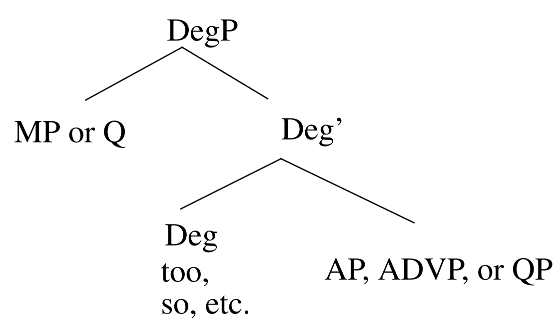
b. six inches too long

c. a dozen fewer books

Because these DP's have such a specific structure and because they closely modify the degree word, I have implemented mensural phrases as "MP's" in a separate network. They are allowed to occur in the specifier position of the DEG phrase, meaning that if the DEG is not lexicalized, the DEGP has an empty head. As was observed previously, this is not particularly difficult because the DEG head is often empty in the case of simple adjectives and adverbs.

Thus, the overall structure that I have implemented as a DEG phrase (DEGP) is like:

(51)



This single structure accounts for adjective, adverb, and quantifier phrases including all of their degree modification. This implementation facilitates modifying the grammar because the kinds of phrases that specify quality, quantity, and description are unified into one structure. Degree Phrases can occur as adjuncts to N' in DP's, as the specifier of PP's, and as adjuncts in VP's.

3.3.4 Adjective Phrases

In addition to this structural account of adjective phrases as complements of degree phrases, I have implemented a preference for adjective ordering. In this way I attempt to describe the scope particular adjectives have over others and explain why in the data below (52a) seems "better formed" than any of (52b-f).

(52) a. rich white American man

b. ??white rich American man

c. ??American rich white man

d. ??rich American white man

e. ??white American rich man

f. ??American white rich man

Based on the work of Quirk and Bache, I have distinguished three types of adjectives (Quirk, 1985), (Bache, 1978) which occur in a particular order. I will discuss the specifics of this implementation in Chapter 4, as the distinctions are encoded as part of the lexicon entry of the adjective. With regard to implementation as Degree Phrases, each adjective is part of its own degree phrase so that the possibility of data like the following can be accounted for:

(53) The [DEGP six feet too [ADJ long]], [DEGP five feet

too [ADJ wide] table.[25]

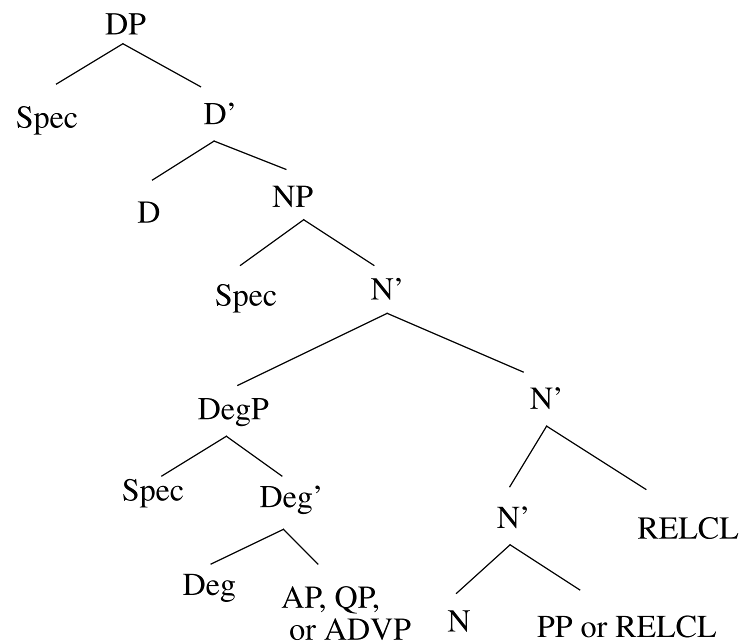
I have implemented this ordering by assigning a number to each type of adjective. When an adjective degree phrase is encountered, if its complement adjective is not of the same number or larger than the number of any previous adjective degree phrases, the adjective sequence will be considered ungrammatical and the sentence will be not parse.

3.3.5 Noun Phrases

The implementation of the Determiner Phrase and Degree phrase I have described above accounts for many structures normally thought of as part of noun phrases. This is a side effect of this version of X' theory, which considers the noun phrase to be a complement of the determiner phrase (cf. structure (35)). Nevertheless, the noun phrase is still a full maximal projection which has a specifier and complement of its own. As expected, the head of the noun phrase is a noun, and this head can have either prepositional phrases or relative clauses as complements. Restrictive relative clauses can also serve as adjuncts to N, as will be discussed in the section on relative clauses below (cf. 3.3.5.2). As I mentioned in the discussion of degree phrases, the specifier position of the NP can hold a quantifier, but is more often an empty position. The noun phrase fits into the overall structure of the determiner phrase like in the tree below:

25. The grammar allows no punctuation so that a phrase like "six feet too long five feet too wide" may actually occur.

(54)

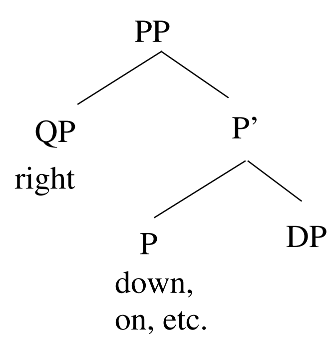


In the following sections I will discuss the parts of this structure in more detail.

3.3.5.1 Prepositional Phrases

Prepositional phrases have a straightforward implementation as shown in the structure below:

(55)



The specifier position holds a Quantifier Phrase, which as discussed previously, is implemented as a Degree Phrase with a quantifier complement. The head is a preposition and the complement a Determiner phrase, which itself could contain prepositional phrases.

3.3.5.2 Relative Clauses

Relative clause implementation is a little more tricky. Relative clauses are CP's (i.e., a sentence in the X' notation) that are either introduced by "that" or a wh-pronoun, which is the head of the CP, or not introduced, in which case the CP has an empty head. The tricky part is that the CP is missing a determiner phrase, or other phrase (e.g., prepositional phrase), usually either in subject or object position. This missing phrase is the one containing the noun that the relative clause is modifying.

My implementation captures this relationship between the relative clause and the moved phrase by putting a copy of the moved noun head back into its original position in the relative clause. This copied noun serves as a kind of "trace" in the noun's original position and maintains number and reference in the relative clause.[26] Agreement occurs between this trace noun just as it would with a normal noun.

The schematic tree structure in (54) shows that, following the analysis given in (Radford, 1988), relative clauses can appear in two places in the noun phrase. This is to account for the differences seen in the following noun phrases, which are taken from (Radford, 1988, p. 218):

---

26. The noun head rather than the entire DP is sufficient as a trace because the noun is all that is necessary to maintain number and reference. It is used because it is the most accessible structure at the point in the parse where relative clauses have been encountered (i.e., a DP has not been completed because the relative clause is part of it and so a full DP is not available to move back to its original position.). The use of the word "trace" here is not equated with any of the traces in GB (i.e., NP-trace or WH-trace), and is only meant to recall that notion.

(56) a. the claim [CP [COMP that] you made a mistake]

b. *the claim [CP [COMP which] you made a mistake]

c. *the claim [CP [COMP 0] you made a mistake]

d. the claim [CP [COMP that] you made]

e. the claim [CP [COMP which] you made]

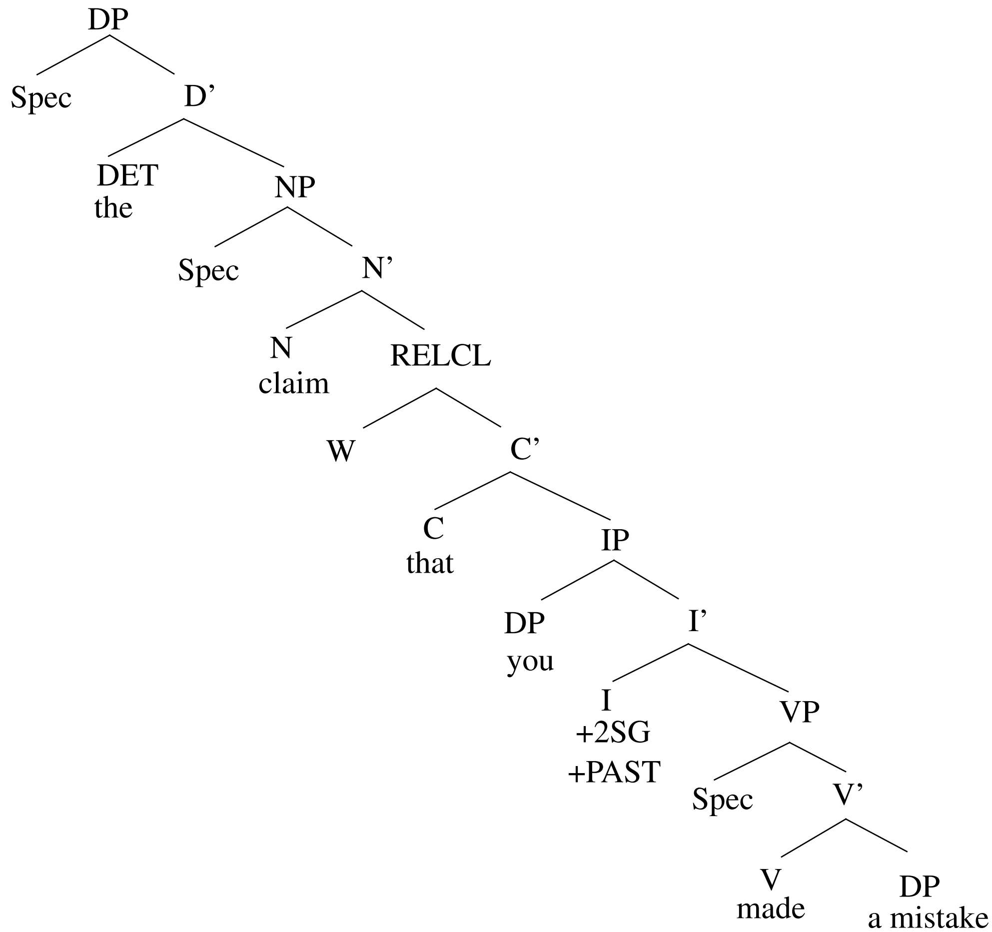
f. the claim [CP [COMP 0] you made]

In this example, the NP's in (56a-c) are "Noun Complement Clauses" which occur as complements to the noun head. They require the complementizer "that" to introduce them and can be introduced by no other relative pronoun. Conversely, it is evident in (56d-f) that these noun phrases are grammatical regardless of what, or if any, relative pronoun introduces them. These relative clauses are called "Restrictive Relative Clauses" and serve only to give extra information about the noun. They are therefore in adjunct position in the overall noun phrase structure.

Thus, a noun phrase with a noun complement relative clause like that in (56a) will have the structure in (57a) in my implementation:

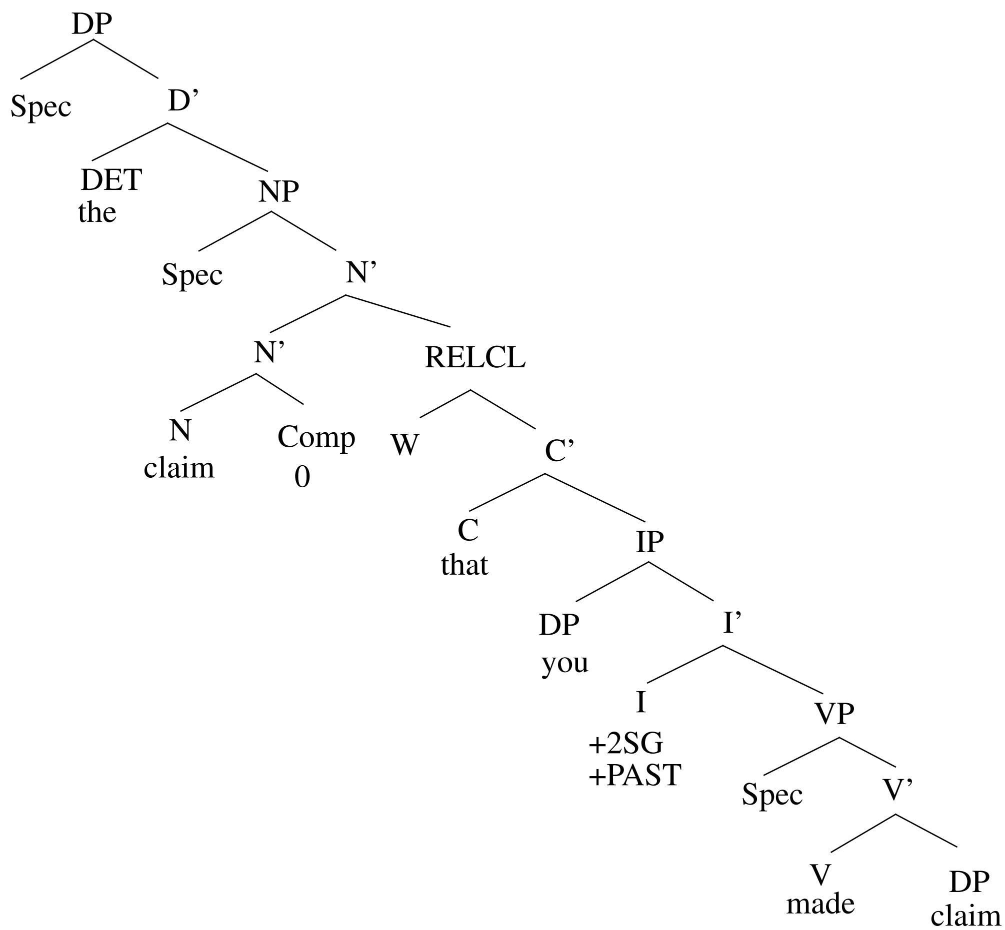
(56a) The claim that you made a mistake.

(57a)



For a noun phrase with a restrictive relative clause like that in (56d), my grammar will produce the structure in (57b):

(57b)



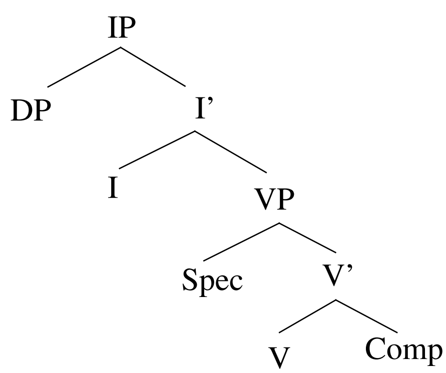
In practice, the grammar will produce both structures for all relative clauses having the "that" complementizer and other factors like semantics must be applied to choose the correct interpretation. For structures with relative pronouns in them, the grammar will only produce structures like that in (57b).

The most important thing to note is that to analyze the relative clauses, this grammar uses the same structure and implementation of a main clause CP like I discussed in Section 3.3.1. Therefore anything that occurs at the level of the main clause, like for example Degree Phrases, can also be accounted for in relative clauses. The only difference is that when the parsing reaches the place in the sentence where the noun is missing, it puts in the noun copy and continues through the parse. This allows for a great economy of structure and accounts for all possible variation within the constituents of relative clauses.

3.3.6 Verb Phrases and Complementation

Under the current X' theory interpretation of sentence structure, VP is the complement of the INFL head. INFL dictates the main verb's person, number, and tense but the main verb is still the head of its own maximal projection. The relationship between INFL and the verb phrase is shown below:

(58)



While syntacticians are not convinced about what structure actually occurs in the specifier position of the VP, based on the data given below, I have implemented an optional ADVP (i.e., DEGP with ADVP complement) in this position.

(59) a. John [DEGP [DEG 0] [ADVP quickly]] ran down the street.

b. Jane was [DEGP [DEG so] [ADVP completely]] exhausted

that he could barely walk.

Adverb degree phrases have also been implemented as adjuncts on V', as have prepositional phrases, finite clauses, and particle words such as "up", or "away".[27] These are licensed to occur with particular kinds of intransitive verbs, since because they are adjuncts, the verb does not subcategorize for them to be in argument positions.

Recall that the head of the determiner phrase selects for an NP complement and that the head of the degree phrase selects for an adjective phrase, adverb phrase, or quantifier phrase complement. In the same way, the head of the verb phrase subcategorizes for its complement. Here, the range of possible complements is much greater and when a particular kind of complement can occur depends on the verb itself. For example the verb "believe" can be followed by a full sentence as in

(60) John believes Mary is sleeping.

but with the verb "take", this structure is ungrammatical:

(61) *John takes Mary is sleeping.

Instead, "take" needs a single noun phrase object and perhaps a prepositional phrase following it, such as:

(62) John takes Mary to the store.

Conversely, the verb "believe" can not have this structure, but can also have a single noun phrase object such as:

---

27. Particles are taken to be bare adverbs and therefore do not adhere to Chomsky's Maximality constraint in the same way that my implementation of specifiers does not. Refer to footnote 13 for a further discussion of this.

(63) John believes Mary.

My implementation accounts for the different kinds of verbs and the different complements they can take with codes in the dictionary entry for each verb. The codes are based on the verb pattern codes in the Oxford Advanced Learners Dictionary (OALD) (Cowie, 1989). The details of these codes are explained in Chapter 4, but here I will discuss the types of complements they can specify.

There are six types of verb complements which occur in various combinations according to the number of arguments a verb subcategorizes for. These are the adjective phrase (DEGP with AP complement), determiner phrase (DP), prepositional phrase (PP), a sentential phrase (CP), small clause (SC), and exceptional clause (EC).

Adjective phrases as complements occur primarily with linking, or copular, verbs such as

(64) a. John is intelligent.

b. The sky became dark.

Determiner phrases may also occur with copular verbs like in (65a), but are most common as direct objects such as (65b-c):

(65) a. John is a farmer.

b. The dog eats his food.

c. The man hit the ball.

Prepositional phrases are usually adjuncts to verb phrase as in (66a), but they can also occur as objects, as in (66b):

(66) a. The man was crying in the living room.

b. The meeting lasted for two hours.

In (66b) the prepositional phrase is a complement rather than an adjunct because the sentence “The meeting lasted” is ungrammatical without it (cf. “The man was crying”). It is evident that the verb “lasted” requires a complement because of sentences like “The meeting lasted a week.”

Sentential complements such as that in (60) take an entire CP as their complement even though there is no complementizer introducing the embedded clause “Mary is sleeping” in (60). Other examples of this complement are sentences like

(67) a. Jane thought [that Mary would take care of her].[28]

b. The man hoped the train would come on schedule.

c. The man hoped that the train would come on schedule.

The implementation of this is simply to allow a CP to be PUSHed for in the complement position. Because it is a full CP, all of the structures possible in the main clause (i.e., degree phrases, relative clauses) are also possible in the complement clause.

Small and Exceptional Clauses, only appear in complement positions and have therefore not been mentioned previously. They lack elements that are part of ordinary

28. According to GB, this example shows that “that Mary would take care of her” is a clause separate from the main clause because “her” is a pronoun and not a reflexive such as in the ungrammatical sentence “*Jane$_i$ thought that Mary would take care of herself$_i$.

CP's: for example a small clause does not have tense because it does not have an INFL node and therefore can not independently constitute a sentence.

Small Clauses (SC) also do not have a Complementizer node, meaning that they can not be introduced with words like "that" and can not serve as relative clauses because there is no structural position for the relative pronoun. Instead they are of the form [DP XP] where XP is any of the other phrasal possibilities (i.e., DP, DEGP, VP, and PP). Examples of Small Clause complements, taken from (Radford, 1988, p. 324), are given below:

(68) a. I believe [the President incapable of deception.] (DP DEGP)

b. I consider [John extremely intelligent.] (DP DEGP)

c. They want [Zola off the team.] (DP PP)

d. Could you let [the cat into the house.] (DP PP)

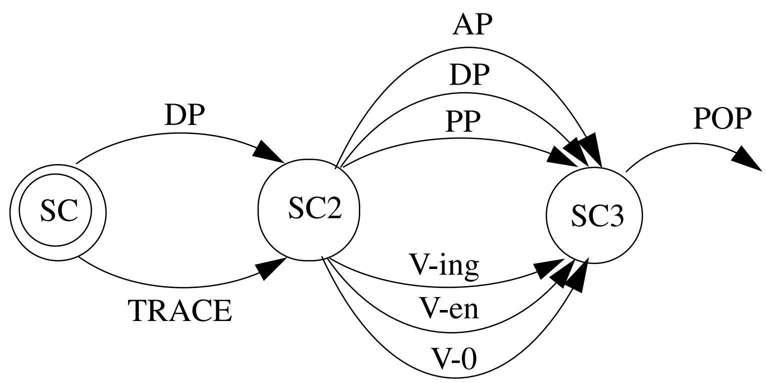
e. Most people find [Syntax a real drag.] (DP DP)

f. Why not let [everyone go home.] (DP VP)

There is sufficient evidence in the syntax literature showing that these structures are in fact clauses rather than a sequence of different complements (cf (Radford, 1988, p. 324-331) and references there). I will not go into this here except to stress that there is a difference between Small clause structures and structures with multiple objects. This becomes clear with verbs that allow single complements versus those that allow more than one. The Small clause is a single constituent and therefore can account for one role in the sentence (i.e., object, direct object, location, etc.). If a verb allows more than one complement to account for different roles, as in a verb that takes both a direct and an indirect object, the small clause could only fill one of these roles.

I have implemented Small Clauses in a separate network of the form [DP XP] where the XP can be a DEGP, a DP, a PP, or one of 3 kinds of VP's: gerundive V-ing forms, participial V-en forms, or infinitival V-0 forms. The network has the structure:

(69)



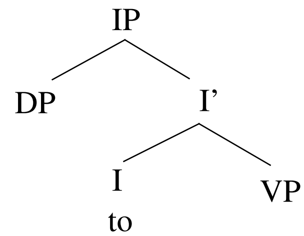
It is possible for the subject DP to be either overt or covert, in which case I will fill this position with a "TRACE" marker, indicating that it is lexicalized elsewhere in the sentence.[29]

Exceptional Clauses also differ from ordinary CP's, as they lack the Complementizer position. They do have an INFL node, but it must always contain "to" and therefore requires the verb to have an infinitival head. Consequently, their basic structure is of the form:

---

29. This analysis accounts for the structure derived by subject raising even though the actual process of raising is not implemented. This structure is only possible with verbs having the code CNI. See Chapter 4 for details.

(70)



The verbs which normally take EC's as complements are usually "cognitive" verbs, such as those shown in (Radford, 1988, p. 317):

(71) a. I believe [the President to be right.]

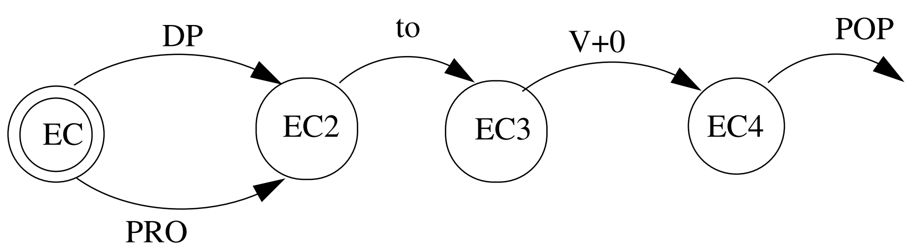
b. I've never known [the Prime Minister to lie.]

c. They reported [the patient to be in great pain].

d. I consider [my students to be conscientious.]

Exceptional Clauses have been implemented as a separate network of the form [DP to VP] where the VP is infinitival and the DP can either be overt or the covert DP called "PRO".[30] The network has the form:

(72)



30. "Big PRO," as PRO is called, has a specific meaning and distribution in GB. This meaning is not pertinent to this project except to say that it is possible to have a PRO subject in an EC because the verb is always infinitival.

As I mentioned previously, when these complements occur depends on the particular verb in the sentence and the code it has in its lexicon. The lexicon is therefore crucial to determining the structure of a sentence. This is predicted in Government and Binding Theory by the Projection Principle, which states that "representations at each syntactic level are projected from the lexicon, in that they observe the subcategorization properties of lexical items. (Sells, 1985)."

I will discuss this in more detail in Chapter 5, but would like to stress that this "lexical determinism" should in no way be considered a problem of the implementation. It does make using the grammar for alternate computational applications somewhat demanding on the computational environment because the lexical entries must be tailored as described in Chapter 4. This is not unexpected, because it is exactly this type of dependence that the linguistic theory predicts.[31]

3.4 Example Parse Trees

This section contains parse trees showing full expansions of the structure the grammar provides for determiner phrases and sentences. The do not show the different types of verb complements the grammar can handle because these are given in Chapter 4. The following parses are actual output from the system.

---

31. I would also like to note that of the three syntactic theories that use X' Theory (i.e., Generalized Phrase Structure Grammar (GPSG), Lexical-Functional Grammar (LFG), and Government and Binding Theory (GB)), GB demands the least amount of information of its lexicon. This is largely because it uses functional heads to build up structure according to X' theory.

(73) Fully expanded Determiner phrase without adjuncts or relative clause:

```
(DP (SPEC ALL)
   (DBAR (HEAD THE)
        (COMP (NP (SPEC MANY)
                 (NBAR (HEAD MEN) (NU (PL)))))))
```

All the many men

```
(DP (SPEC ALL)
   (DBAR (HEAD THE)
        (COMP
          (NP (SPEC (MP (SPEC TWO)
                       (MBAR (HEAD WEEK)
                            (COMP NIL))))
             (NBAR (HEAD VACATIONS)
                  (NU (PL)))))))
```

All the two week vacations

(74) Determiner phrases with Degree phrase adjunct:

```
(DP (SPEC ALL)
   (DBAR (HEAD THE)
        (COMP
          (NP (SPEC NIL)
             (NBAR
               (ATTRIB
                 (DEGP
                   (SPEC (MP (SPEC TWO)
                            (MBAR (HEAD WEEK)
                                 (COMP NIL))))
                   (DEGBAR (HEAD TOO)
                          (COMP (AP (SPEC NIL)
                                   (ABAR (HEAD LONG)))))))
               (HEAD VACATIONS)
               (NU (PL)))))))
```

All the two week too long vacations

```
(DP (SPEC ALL)
   (DBAR (HEAD THE)
        (COMP
          (NP (SPEC (MP (SPEC TWO)
                        (MBAR (HEAD WEEK)
                              (COMP NIL))))
              (NBAR
                (ATTRIB
                  (DEGP (SPEC NIL)
                        (DEGBAR (HEAD NIL)
                                (COMP (AP (SPEC NIL)
                                          (ABAR (HEAD LONG)))))))
                (HEAD VACATIONS)
                (NU (PL)))))))
```

```
(DP (SPEC ALL)
   (DBAR (HEAD THE)
        (COMP
          (NP (SPEC NIL)
              (NBAR
                (ATTRIB
                  (DEGP
                    (SPEC (MP (SPEC TWO)
                              (MBAR (HEAD WEEK)
                                    (COMP NIL))))
                    (DEGBAR (HEAD NIL)

                            (COMP (AP (SPEC NIL)
                                      (ABAR (HEAD LONG)))))))
                (HEAD VACATIONS)
                (NU (PL)))))))
```

All the two week long vacations

```
(DP (SPEC NIL)
   (DBAR (HEAD NIL)
        (COMP
          (NP (SPEC NIL)
              (NBAR
                (ATTRIB
                  (DEGP (SPEC MUCH)
                        (DEGBAR (HEAD TOO)
                                (COMP (AP (SPEC NIL)
                                          (ABAR (HEAD MUCH)))))))
                (HEAD HAIR)
                (NU (SG/PL))))))
```

Much too much hair

(75) Determiner phrase with restrictive relative clause adjunct:

```
(DP (SPEC NIL)
  (DBAR (HEAD THE)
     (COMP
      (NP (SPEC NIL)
        (NBAR (HEAD MAN)
           (NU (SG))
           (RESTRICT
            (REL-CLAUSE
             (CP (SPEC WHO)
               (IP
                (SPEC
                 (DP (SPEC NIL)
                   (DBAR (HEAD NIL)
                      (COMP
                       (NP (SPEC NIL)
                         (NBAR (HEAD MAN)
                            (NU (SG))))))))
                (IBAR (HEAD (AGR 3SGPRES))
                   (COMP
                    (VP (SPEC NIL)
                      (VBAR (HEAD HATE)
                         (COMP
                          (OBJ
                           (DP (SPEC NIL)
                             (DBAR (HEAD THE)
                                (COMP
                                 (NP
                                  (SPEC NIL)
                                  (NBAR
                                   (HEAD
                                   WOMAN)
                                   (NU (SG))))))))))))))))))))))
The man who hates the woman
```

(76) Determiner phrase with relative clause noun complement:

```
(DP (SPEC NIL)
   (DBAR (HEAD THE)
        (COMP
          (NP (SPEC NIL)
             (NBAR (HEAD GAMES)
                  (NU (PL))
                  (COMP
                    (REL-CLAUSE
                      (CP (SPEC THAT)
                         (IP
                           (SPEC
                             (DP (SPEC NIL)
                                (DBAR (HEAD THE)
                                     (COMP
                                       (NP (SPEC NIL)
                                          (NBAR (HEAD BOYS)
                                               (NU (PL)))))))))
                           (IBAR (HEAD (AGR 3PLPRES))
                                (COMP
                                  (VP (SPEC NIL)
                                     (VBAR (HEAD PLAY)
                                          (COMP
                                            (OBJ
                                              (NBAR (HEAD GAMES)
                                                   (NU (PL))))))))))))))))))))
```
[32]

32. Transitive verb generates the “games” trace in object position.

```
(DP (SPEC NIL)
    (DBAR (HEAD THE)
          (COMP
            (NP (SPEC NIL)
                (NBAR (HEAD GAMES)
                      (NU (PL))
                      (COMP
                        (REL-CLAUSE
                          (CP (SPEC THAT)
                              (IP
                                (SPEC
                                  (DP (SPEC NIL)
                                      (DBAR (HEAD THE)
                                            (COMP
                                              (NP (SPEC NIL)
                                                  (NBAR (HEAD BOYS)
                                                        (NU (PL))))))))
                                (IBAR (HEAD (AGR 3PLPRES))
                                      (COMP
                                        (VP (SPEC NIL)
                                            (VBAR (HEAD PLAY))))))))))))))[33]
```

The games that the boys play

(77) Fully expanded Sentence (CP) with non-lexicalized INFL:

```
(DCL
  (CP (SPEC NIL)
      (CBAR (HEAD NIL)
            (COMP
              (IP
                (SPEC
                  (DP (SPEC NIL)
                      (DBAR (HEAD THE)
                            (COMP
                              (NP (SPEC NIL)
                                  (NBAR (HEAD BOYS)
                                        (NU (PL))))))))
                (IBAR (HEAD (AGR 3PLPRES))
                      (COMP (VP (SPEC NIL)
                                (VBAR (HEAD PLAY))))))))))
```

The boys play

33. Play is also an intransitive verb and therefore has no object in this parse.

(78) Sentences with different INFL's:

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP
        (SPEC
         (DP (SPEC NIL)
           (DBAR (HEAD THE)
              (COMP
               (NP (SPEC NIL)
                 (NBAR (HEAD BOY)
                    (NU (SG))))))))
        (IBAR (HEAD (AGR 3SGPRES)
              (BE ING))
           (COMP (VP (SPEC NIL)
                (VBAR (HEAD PLAY))))))))))
```

The boy is playing

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP
        (SPEC
         (DP (SPEC NIL)
           (DBAR (HEAD THE)
              (COMP
              (NP (SPEC NIL)
                 (NBAR (HEAD BOY)
                    (NU (SG))))))))
        (IBAR (HEAD (AGR 3SGPRES)
              (HAVE EN)
              (BE ING))
           (COMP (VP (SPEC NIL)
                (VBAR (HEAD PLAY))))))))))
```

The boy has been playing

```
(DCL
 (CP (SPEC NIL)
    (CBAR (HEAD NIL)
         (COMP
           (IP
             (SPEC
               (DP (SPEC NIL)
                   (DBAR (HEAD THE)
                         (COMP
                           (NP (SPEC NIL)
                               (NBAR (HEAD BOY)
                                     (NU (SG))))))))
             (IBAR
               (HEAD (NEG NOT)
                     (AGR 3SGPAST)
                     (MODAL CAN)
                     (HAVE EN)
                     (BE ING))
               (COMP (VP (SPEC NIL)
                         (VBAR (HEAD PLAY))))))))))
```

The boy could not have been playing.

Chapter 4

THE LEXICON

This chapter contains specific descriptions of the subcategorization codes and other syntactic information that appears in the lexical entry of each word. It is intended to help someone new to this system add new entries into the system's dictionary. Here I indicate the lexical coding the grammar requires and discuss how these codes might be derived from an already tagged corpus. This chapter also gives the reader a good feeling for the coverage of the grammar provided with the predictor. The full working dictionary is included in the appendices.

4.1 Sources and Considerations

A major concern for building this dictionary is to enable a person not familiar with the implementation of the grammar to easily construct lexical entries. My first step toward this end has been to adopt traditional word categories of the sort explained in the introduction to any good desk dictionary. These will include determiners (of which articles are a subset), nouns, adjectives, verbs, adverbs, prepositions, pronouns, and conjunctions. This allows the word categories to be non-specific to this project and permits the grammar to share on-line dictionaries already available. All that would be required for this system is that the additional features described here be added into the lexical entries. This addition is considered less demanding than a complete recategorization of each word in the lexicon.

The features on each word hold the specifications for how that word can be used and what structures can follow it. In the case of nouns and verbs, they were devised based on the coding in the Oxford Advanced Learner's Dictionary (OALD) (Cowie, 1989). Attention was also paid to the word tags that might be found in a large corpus of English, such as the Brown Corpus (Francis and Kucera, 1982). This corpus contains approximately 1,000,000 words tagged with combinations of 87 word "tags." These tags typically give the word's syntactic category, but other usage information can be included. For example, the word "Great" in "Alexander the Great's" would be tagged with the adjective tag "JJ" plus the possessive tag "$". There are also tags to denote foreign words, "FW", cited words, "NC", and words appearing in titles, "TL."

The Brown Corpus of American English was the earliest tagged corpus, and for this reason most other corpora tagsets are derived from Brown's. There are drawbacks to using the Brown tagset which stem mainly from the fact that it does not sufficiently differentiate certain kinds of words. For example, the Brown corpus only has four basic tags for nouns: "NN" for singular common nouns, "NNS" for plural common nouns, "NP" for proper nouns, and "NR" for adverbial nouns. With this system there is no way to distinguish mass nouns and this is information necessary for determiner subcategorization (i.e., to rule out sentences such as "*a furniture arrived" and "boy walked"). Tus, there is not a direct correspondence between the codes used in this project and the Brown tags. Nonetheless, when appropriate I will note what Brown corpus tags correspond to the lexicon features used here. This will facilitate automatic translation of corpus tags to the codes needed here because the tags can serve as a starting point in the process. It will be necessary to manually check each word tagged with the Brown tags to check and further categorize the word; however, this has been viewed as a critical process in many automatic text tagging projects (Keulen, 1986), (Marcus et al., 1990).

One of several alternate tagsets, also based on Brown's tags, was devised to assign categories to the LOB corpus of British English. This tagset is significantly more descriptive than the Brown set; it not only differentiates common and mass nouns, but also differentiates mensural, proper, and titular nouns, more clearly specifies the pronouns, separates relative and wh-words, and identifies negative words. In addition there is a Lancaster tagset which was devised specifically for parsing and therefore makes very fine distinctions between words. Of course as these tagsets become more descriptive, they are also larger and more complicated to use. The fact that they are based on the Brown tagset should make my Brown tag notations helpful for using these corpora.

A final attempt to facilitate using this system with an already existing dictionary is that the values assigned to the properties in the dictionary, such as (SG) as the argument to the NUMBER property of a noun), have been constructed to work with macros. This would allow the numbering system used in a pre-existing dictionary to be hidden from the grammar: the grammar would access the macro rather than the direct number information. The surface value will be what the grammar calls for, but the internal specifications can come from the new dictionary. Exploiting this capability may require some extra macro programming, but will allow for maximal compatibility between systems.

4.2 Implementation

The categories used to create the dictionary for this grammar were taken from the American Heritage Dictionary, Second College Edition (Revised edition, 1976). This dictionary was used as representative of the generally accepted categorial interpretations of words. Often the categories found here are less specific than those found in other dictionaries, but the objective in using them was to preserve main-stream word interpretations.

For the verb subcategorization information I have gone to a more specialized dictionary: the Oxford Advanced Learner's Dictionary (OALD) (Cowie, 1989). The OALD divides verbs into five types: linking, intransitive, transitive, complex-transitive, and di-transitive. In the section below I will describe each of these verb types and provide actual parses from the system to illustrate the differences in the structures they produce.

A linking verb functions as an equivalence which assigns characteristics described in the complement to the subject. There are two kinds of syntactic structures produced by these verbs depending on whether they take adjective or noun complements. The predicate nominative construction is illustrated below in (1), while the predicate adjective construction is illustrated in (2):

(1)

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP
        (SPEC
         (DP (SPEC NIL)
           (DBAR (HEAD THE)
              (COMP
               (NP (SPEC NIL)
                 (NBAR (HEAD BOY)
                    (NU (SG))))))))
        (IBAR (HEAD (AGR 3SGPRES))
           (COMP
            (VP (SPEC NIL)
              (VBAR (HEAD BE)
                 (COMP
                  (SUBJECT_COMPLEMENT
                   (DEGP (SPEC NIL)
                      (DEGBAR (HEAD NIL)
                          (COMP
                           (AP (SPEC NIL)
                              (ABAR (HEAD SICK)))))))))))))))))
```

The boy is sick

(2)

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
        (IP
          (SPEC
           (DP (SPEC NIL)
             (DBAR (HEAD THE)
                (COMP
                  (NP (SPEC NIL)
                    (NBAR (HEAD MAN)
                       (NU (SG))))))))
          (IBAR (HEAD (AGR 3SGPRES))
             (COMP
               (VP (SPEC NIL)
                 (VBAR (HEAD BE)
                    (COMP
                     (SUBJECT_COMPLEMENT
                      (DP (SPEC NIL)
                        (DBAR (HEAD A)
                           (COMP
                             (NP (SPEC NIL)
                               (NBAR (HEAD TEACHER)
                                  (NU (SG)))))))))))))))))))
The man is a teacher
```

Intransitive verbs do not subcategorize for complements, although they can have different kinds of adjuncts. The verb codes determine what kinds of adjuncts these verbs can occur with: either a PP, a DEGP, a DP, or an EC. Because they are adjuncts, none of these structures are necessary for forming grammatical sentences and a single verb will very often have codes for a number of these adjuncts. The following structures are produced by the system; the parse in (3) shows a simple intransitive verb with no adjunct, and the parse in (4) shown a prepositional phrase adjunct.

(3)

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP
        (SPEC
         (DP (SPEC ALL)
           (DBAR (HEAD THE)
              (COMP
               (NP (SPEC NIL)
                 (NBAR (HEAD STUDENTS)
                    (NU (PL))))))))
        (IBAR (HEAD (AGR 3PLPAST))
           (COMP (VP (SPEC NIL)
                (VBAR (HEAD CHATTER))))))))))
```

All the students chattered

(4)

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP
        (SPEC
         (DP (SPEC NIL)
           (DBAR (HEAD NIL)
              (COMP
               (NP (SPEC MANY)
                 (NBAR (HEAD STUDENTS)
                    (NU (PL))))))))
        (IBAR (HEAD (AGR 3PLPAST))
           (COMP
            (VP (SPEC NIL)
              (VBAR (HEAD CHATTER)
                 (ADJUNCT
                  (PP (SPEC NIL)
                    (PBAR (HEAD ABOUT)
                       (COMP
                        (DP (SPEC NIL)
                          (DBAR (HEAD THE)
                             (COMP
                              (NP (SPEC NIL)
                                (NBAR
                                 (HEAD EXAM)
                                 (NU (SG))))))))))))))))))))
```

Many students chattered about the exam

Transitive verbs are considered to be two-place predicates, taking a subject and a direct object. The most typical object is a determiner phrase, but other structures that can serve as objects include small clauses of various kinds, ordinary CP clauses whose COMP node is often filled with “that,” and Exceptional clauses. Below in (5) is an example of a structure with a DP object, and (6) shows an example of a CP object:

(5)

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP
        (SPEC
         (DP (SPEC NIL)
           (DBAR (HEAD THE)
              (COMP
               (NP (SPEC NIL)
                 (NBAR (HEAD BOY)
                    (NU (SG))))))))
        (IBAR (HEAD (AGR 3SGPAST))
           (COMP
            (VP (SPEC NIL)
              (VBAR (HEAD HIT)
                 (COMP
                  (OBJ
                   (DP (SPEC NIL)
                     (DBAR (HEAD THE)
                        (COMP
                         (NP (SPEC NIL)
                           (NBAR (HEAD BALL)
                              (NU (SG))))))))))))))))))))
```

The boy hit the ball

(6)

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP
        (SPEC
         (DP (SPEC NIL)
           (DBAR (HEAD THE)
              (COMP
               (NP (SPEC NIL)
                 (NBAR (HEAD TEACHER)
                    (NU (SG))))))))
        (IBAR (HEAD (AGR 3SGPAST))
           (COMP
            (VP (SPEC NIL)
              (VBAR (HEAD BELIEVE)
                 (COMP
                  (OBJ
                   (DCL
                    (CP (SPEC NIL)
                      (CBAR (HEAD (COMPLEMENTIZER THAT))
                         (COMP
                          (IP
                           (SPEC
                            (DP (SPEC NIL)
                              (DBAR (HEAD THE)
                                 (COMP
                                  (NP (SPEC NIL)
                                   (NBAR (HEAD EXAM)
                                   (NU (SG))))))))
                           (IBAR
                            (HEAD (AGR 3SGPAST))
                            (COMP
                             (VP (SPEC NIL)
                               (VBAR (HEAD BE)
                                  (COMP
                                   (SUBJECT_COMPLEMENT
                                     (DEGP (SPEC NIL)
                                       (DEGBAR (HEAD NIL)
                                         (COMP
                                          (AP (SPEC NIL)
                                           (ABAR (HEAD DIFFICULT))))))))))))

                      ))))))))))))))
```

Complex-Transitive is the name the OALD gives to verbs that are three-place predicates. These verbs have a primary DP object and a secondary object which modifies the primary object. The secondary object could take the form of a verb phrase with the verb ending in -ing, an infinitival verb phrase, determiner phrases, adjective phrases, and exceptional clauses. Government and Binding syntax analyzes the structures these verbs produce in terms of small clauses and this is how they have been implemented here. In the examples below, I show an adjective degree phrase object complement in (7), an exceptional clause object complement in (8), and an infinitival small clause object complement in (9).[1]

---

1. Notice in examples (8) and (9) the subject of the EC and SC is PRO and TRACE, respectively. The structure in (8) accounts for the analysis GB gives to Object-Controlled Equi and Subject-to-Object Raising constructions. GB predicts that these are derived through different mechanisms, but their surface structures are similar and I have therefore accounted for them similarly. Refer to Chapter 5 for further discussion of this issue.

(7)

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP
        (SPEC
         (DP (SPEC NIL)
           (DBAR (HEAD THE)
              (COMP
               (NP (SPEC NIL)
                 (NBAR (HEAD TEACHER)
                    (NU (SG))))))))
        (IBAR (HEAD (AGR 3SGPRES))
           (COMP
            (VP (SPEC NIL)
              (VBAR (HEAD KEEP)
                 (COMP
                  (OBJ
                   (SMALL_CLAUSE
                    (DP (SPEC NIL)
                      (DBAR (HEAD THE)
                         (COMP
                          (NP (SPEC NIL)
                            (NBAR (HEAD STUDENTS)
                               (NU (PL)))))))
                    (DEGP (SPEC NIL)
                       (DEGBAR (HEAD NIL)
                           (COMP
                            (AP (SPEC NIL)
                              (ABAR (HEAD BUSY))))))))))))))))))))
```

The teacher keeps the students busy

(8)

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP
        (SPEC
         (DP (SPEC NIL)
           (DBAR (HEAD THE)
              (COMP
               (NP (SPEC NIL)
                 (NBAR (HEAD MAN)
                    (NU (SG))))))))
        (IBAR (HEAD (AGR 3SGPAST))
           (COMP
            (VP (SPEC NIL)
              (VBAR (HEAD FORCE)
                 (COMP
                  (OBJ
                   (DP (SPEC NIL)
                     (DBAR (HEAD THE)
                        (COMP
                         (NP (SPEC NIL)
                           (NBAR (HEAD WOMAN)
                              (NU (SG)))))))
                   (OBJ_COMP
                    (EXCEPTIONAL_CLAUSE
                     (DP (DBAR PRO))
                      (INFL TO)
                       (VP (SPEC NIL)
                        (VBAR (HEAD GIVE)
                           (COMP
                            (IOBJ
                             (DP (SPEC NIL)
                               (DBAR (HEAD (PRO HIM))
                                 (NU (SG))
                               (COMP NIL)))
                                (OBJ
                                  (DP (SPEC NIL)
                                (DBAR (HEAD (PRO HER))
                                 (COMP (NP (SPEC NIL)
                                   (NBAR
                                   (HEAD MONEY)
                                   (NU (SG)))))))))))))))))))))))))))
```

The man forced the woman to give him her money

(9)

```
(DCL
 (CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP
        (SPEC
         (DP (SPEC NIL)
           (DBAR (HEAD THE)
              (COMP
               (NP (SPEC NIL)
                 (NBAR (HEAD TEACHER)
                    (NU (SG))))))))
        (IBAR (HEAD (AGR 3SGPRES))
           (COMP
            (VP (SPEC NIL)
              (VBAR (HEAD LET)
                 (COMP
                  (OBJ
                   (DP (SPEC NIL)
                     (DBAR (HEAD (PRO HER))
                        (COMP
                         (NP (SPEC NIL)
                           (NBAR (HEAD STUDENTS)
                              (NU (PL)))))))
                   (OBJ_COMP
                    (SMALL_CLAUSE
                      (DP (DBAR TRACE))
                      (VP (SPEC NIL)
                        (VBAR (HEAD PLAY)))))))
                         (ADJUNCT
                          (PP (SPEC NIL)
                            (PBAR (HEAD IN)
                              (COMP
                                (DP (SPEC NIL)
                                 (DBAR (HEAD (PRO HER))
                                     (COMP
                                        (NP (SPEC NIL)
                                      (NBAR (HEAD CLASSES)
                                              (NU (PL))))))))))))))))))))
```

The teacher lets her students play in her classes

Di-transitive verbs are those that take an indirect object, which is either a DP or a PP. Their object can be another DP, a PP, a CP, or an Exceptional Clause. They are different from the Complex-transitive verbs because the two constituents following them have particular semantic roles (i.e., object and indirect object). In the two examples that follow, I have given the most typical D-transitive verb construction in (10) and a construction in (11) having a PP indirect object and a CP object.

(10)

```
(DCL
 (CP (SPEC NIL)
    (CBAR (HEAD NIL)
         (COMP
          (IP
            (SPEC
             (DP (SPEC NIL)
                (DBAR (HEAD THE)
                     (COMP
                      (NP (SPEC NIL)
                         (NBAR (HEAD MAN)
                              (NU (SG))))))))
            (IBAR (HEAD (AGR 3SGPAST))
                 (COMP
                  (VP (SPEC NIL)
                     (VBAR (HEAD TEACH)
                          (COMP
                           (OBJ
                            (DP (SPEC NIL)
                               (DBAR (HEAD NIL)
                                    (COMP
                                     (NP (SPEC NIL)
                                        (NBAR (HEAD ENGLISH)
                                             (NU (SG/PL)))))))
                            (IOBJ
                             (PP (SPEC NIL)
                                (PBAR (HEAD TO)
                                     (COMP
                                      (DP (SPEC NIL)
                                         (DBAR (HEAD THE)
                                              (COMP
                                               (NP (SPEC NIL)
                                                 (NBAR (HEAD STUDENTS)
                                                   (NU (PL))))))))))))))))))))))))
```

The man taught English to the students.

(11)

```
(DCL
(CP (SPEC NIL)
   (CBAR (HEAD NIL)
      (COMP
       (IP (SPEC
         (DP (SPEC NIL)
           (DBAR (HEAD THE)
              (COMP
               (NP (SPEC NIL)
                 (NBAR (HEAD WOMAN)
                    (NU (SG))))))))
        (IBAR (HEAD (AGR 3SGPAST))
           (COMP
            (VP (SPEC NIL)
              (VBAR (HEAD ANNOUNCE)
                 (COMP
                  (IOBJ
                   (PP (SPEC NIL)
                     (PBAR (HEAD TO)
                        (COMP
                         (DP (SPEC NIL)
                           (DBAR (HEAD (PRO HER))
                              (COMP
                               (NP (SPEC NIL)
                                 (NBAR (HEAD GUESTS)
                                  (NU (PL))))))))))
                   (OBJ
                    (DCL
                     (CP (SPEC NIL)
                       (CBAR
                         (HEAD (COMPLEMENTIZER THAT))
                        (COMP
                          (IP (SPEC (DP (SPEC NIL)
                               (DBAR (HEAD NIL)
                                 (COMP (NP (SPEC NIL)
                                   (NBAR (HEAD DINNER)
                                    (NU (SG/PL))))))))
                          (IBAR  (HEAD (AGR 1SGPAST))
                           (COMP
                            (VP (SPEC NIL)
                              (VBAR (HEAD BE)
                                 (COMP
                                  (SUBJECT_COMPLEMENT
                                   (DEGP (SPEC NIL)
                                     (DEGBAR (HEAD NIL)
                                      (COMP (AP (SPEC NIL)
                                        (ABAR (HEAD READY))))))))))  )))))))))))))))))
```

The woman announced to her guests that dinner was ready.

All the codes given in the OALD for that verb are included in a verb's lexical entry so that all structures that could possibly complement the main verb are described. This makes the grammar dependent on the OALD codes and consequently, the information in the lexicon is very specific to this project. The best scenario would be to use Roger Mitton's computerized version of the OALD (see reference in (Garside, et al., 1987), but other electronic dictionaries could be used as well. The verb subcategorization information given in the OALD could be derived from information in the electronic version of Longman's Dictionary of Contemporary English (LDOCE). A problem with using this dictionary is that the subcategorization information is not as accessible as in OALD because there is no formal coding like OALD uses. This can be overcome with an automatic translator that could convert the LDOCE codes, and then it would be possible to use this dictionary as the large dictionary this grammar requires. This procedure could be recreated with most any electronic dictionary or text as long as it includes codes or tags which are sufficiently descriptive.

4.3 Lexical Entries

Each lexical entry has been implemented as a LISP symbol whose form is equivalent to the word itself. Each of these symbols has an associated property list which has the following schematic form:

(12) <CATEGORY X> <toggled features> <value features> <ROOT X>

Here, X is a variable which is equal to either the category or the root of the word, as appropriate. Toggled features are those whose value is either "t" or "nil" and include the verb subcategorization codes. Only the relevant features are included in a word's dictionary entry, meaning that the lexical entry will contain no features with the value "nil". Value features, the most important of which is "NUMBER", have a set value which is specific to that word. These are the features I mentioned previously as being engineered to accept macros as values, but they may also take LISP lists as values. The ROOT property is optional to allow for categories whose lexicalization is static (e.g., prepositions). The ROOT property is most required for verbs, but must also be used with other inflected words, such as comparative and superlative adjectives.

4.3.1 CATEGORY

As explained above, this system primarily uses the traditional category assigned to a word, with the possible exception of the category MODAL which is given to words like "would" and "could". The arguments to the CATEGORY property, along with reference to the appropriate Brown Corpus tags from which they could be derived, are given in Table 4.1 below.

Table 4.1 CATEGORY codes and sources

CODE	TRADITIONAL CATEGORY	BROWN CORPUS TAGS
N	Noun	NN, NNS, NP, NPS,NR, NRS
V	Verb	VB, VBD, VBG, VBN,VBZ
DET	Determiner	ABL, ABN, ABX, AT, WDT
PRO	Pronoun	PN, PN$, PP$, PP$$, PPL, PPLS, PPO, PPS, PPSS, WP$, WPO, WPS,
ADJ	Adjective	JJ, JJR, JJS, JJT, QLP, AP
PREP	Preposition	IN
CONJ	Conjunction	CC, CS
ADV	Adverb	QL, RB, RBR, RBT, RN, RP, WQL, WRB
MODAL	Modal	MD

To create a dictionary entry for a word, it must be assigned at least one category from this list. This is done with an association like “CATEGORY V” in the property list. A word could be assigned multiple categories by making the argument to the “CATEGORY” property a list containing the appropriate codes. For example, the word “garden”, which could be both a verb or noun, would have the category entry “CATEGORY (N V)”.

Each lexical entry must contain all the category codes and lexical features that apply to that word because the grammar only accesses the last word entry read into memory. This means that the lexical entry for “garden” will include the specifications for its noun sense (cf. Noun section below) and also the specifications for its verb sense (cf. Verb section below). It will have the following form:

(13) (setplist ‘garden ‘(CATEGORY (N V) COUNT t UNTENSED t I t IPR t NUMBER (SG) PNCODE (X3SG) ROOT garden))[2]

Notice this entry follows the schematic given in (12): category names, valued features, toggled features, and lastly, root.

4.3.2 Toggled features

These are features that describe when a word can be used and with what other words: these represent the word’s subcategorization. A list of possible toggled features and the word categories they can be used with is given in (2). For further details about each feature refer to the section below on that feature.

Table 4.2 Category Features

CATEGORY	FEATURE
DET	PRE-DET, CENTRAL-DET, QUANT, ART, REL, WH, NP
ADJ	QUANT, CENTRAL-DET, PRE-DET, PRED
N	CARDINAL, MASS, COUNT, PROPER PRE-DET, MENSURAL
PRO	CENTRAL-DET, PROPER, POSS, DEM, PRE_DET, WH, REL
ADV	DEG, NEG, WH, REL
V	CNT, CNG, CNA, CNN, DNPR, DNF, CNI, DNN, DPRW, DNT, DPRF, DNW, LN, I, IPR, IP, INPR, IT, PRESPART, LA, UNTENSED, PASTPART, DPRT, TNP, TF, TNT, TG, TNG, TNI, TN, TNPR, TW, TT

2. The value of PNCODE in this entry is a macro which has the value of the following list: (1SG-PRES 2SGPRES 1PLPRES 2PLPRES 3PLPRES).

All of these features will be assigned the value “t” when they occur in a lexical entry, because they are only included when appropriate. The property assignment takes the form of “<FEATURE> t” (i.e.,“CENTRAL-DET t”). Just as with the categories, all features that apply to any sense or usage of the word must be included in a single lexical entry. This means that it is possible for noun and verb features to occur in the same lexical entry, as was the case in the “garden” entry of (13).

4.3.2.1 PRE-DET

This feature is used to specify that a word can occur in “pre-determiner” position, which means that it can occur in the environment “X the noun.” Examples of predeterminers are “all, half, both, etc.” Also categorized as PRE-DET are multipliers like “twice, thrice, and once” and fractions like “one-third, one-half.” Pre-determiners correspond to the Brown corpus tags ABL, ABN, ABX. They can be adjectives, pronouns, or nouns.

4.3.2.2 CENTRAL-DET

This feature is used to specify words that occur in the central position in a determiner phrase. It includes articles and other words falling into the canonical designation “determiner”. In addition, this feature is appropriate for the possessive pronouns like “my” and “his,” demonstrative pronouns such as “that” and “this,” and quantifying adjectives like “every” and “some.” A word should receive this feature if it can occur in the environment “all X noun” as an instantiation of X. These words correspond to the Brown corpus tags AT, PP$, and DT, DTI, DTS, DTX.

4.3.2.3 ART

This feature further classifies a determiner as an article. It is typically assigned to the words "a", "an", and "the." These words correspond directly to the Brown corpus tag AT.

4.3.2.4 REL

This feature is used to identify words that could be used to introduce restrictive relative clauses. Typically it is given to determiners like "which", pronouns like "who" and "whose", adverbs like "when", and also the word "that." It can be given to any word with the Brown corpus tag WDT. If a word can introduce a relative clause, replacing "X" in the following sentences, it should be given this feature:

(14) a. John X stole the mulberry cake...

b. The dog X ate my homework...

c. The time X I told you...

d. The woman X turtle I borrowed...

4.3.2.5 WH

This feature identifies determiners, adverbs, and pronouns that could introduce WH questions. Its distribution largely overlaps that of the REL feature, but is separated in order to account for words like "that" which are REL but not WH. Words that receive this tag, behave like the following:

(15) a. X stole the mulberry cake?

b. X dog ate the plants?

c. X ate the roots of the plant?

d. X did you discover the plant was missing?

It can be given to most words with the Brown corpus tags WDT, WP$, WPO, WPS, WQL, WRB.

4.3.2.6 NP

This code is to denote which determiners require noun phrases as complements. For example the determiner “the” can not stand alone as the subject of a sentence, but the determiner “that” can. Consider the sentences below:

(16) a. *The is going to the store.

b. *The is funny.

c. This is going to be difficult.

d. That is funny.

If a determiner can occur in sentences like (16c-d) it should be given the feature NP.

4.3.2.7 POSS

This feature further specifies a pronoun as a possessive form and is typically assigned to possessive pronouns like “his” or “her.” It can also be assigned to words with the Brown corpus tags PP$, PP$$, WP$, or any other combination including a “$”. If the word infers ownership, the word should receive this tag.

4.3.2.8 DEM

Specifies a demonstrative pronouns as such. It is given to words like “that” or “this” and some words with the Brown corpus tags DT, DTS. These words can occur in environments such as:

(17) a. X nasty old man...

b. So it was X boy that plucked Annie’s flowers!

c. X is ludicrous.

4.3.2.9 MENSURAL

This feature classifies nouns as quantifying units, for example “half”, “pair”, “dozen”, “bushel”, “mile”, etc. Classification must be done intuitively by considering what kinds of words can be used as a unit of measurement.

4.3.2.10 QUANT

This feature identifies adjectives as quantifiers. It includes cardinal numbers, ordinal numbers, and some specific adjectives like “many”, “few”, “much”, “little”, “several”, “every”, “some”, etc. If the word tells “how much” or “how many” then it should receive this feature. These words typically would have the Brown corpus tags AP, CD, OD, and some of QLP.

4.3.2.11 CARDINAL

This feature is used to separate the cardinal numbers from the quantifiers. It is given to numbers like “one”, “two”, and “thirty”. It corresponds directly to the Brown tag CD.

4.3.2.12 MASS

This feature is assigned to nouns which refer to substances, qualities, collections: objects which cannot be individuated or whose individual parts can not be counted. Examples of these nouns are “anger”, “money”, “oats”, “gasoline”, and “physics”. A useful test for a word like this is to ask “If part of this substance is removed, can I still call what is left by the same word?” For example, if there is a quantity of sugar, and some is removed, what is left can still be called “sugar”; it has not lost its character. In contrast, if the bread is taken away from a sandwich, what is left can not still be called a “sandwich” and so “sandwich” would not be considered a mass noun.

A syntactic test for mass nouns is whether it can occur without a determiner. If it is ungrammatical to say either “X sit” or “X sits” then it is likely that the word X is not a mass noun. These words are usually considered to be singular, although this is not always the case, such as the word “oats” (cf “oats sit” vs. *”oats sits”). In this case the word is given a “neutral” number signifying that it is both singular and plural (e.g., SG/PL). This feature corresponds to the OALD code U or to some words with the Brown Corpus tags NN, NR, NRS.

4.3.2.13 COUNT

This feature is for countable nouns, which contrast with the mass nouns such that if a noun is not a mass noun, it is probably a count noun.[3] Count nouns include words that

3. It is possible for a noun to be both a count and a mass noun. These are called “Dual” nouns and should be given the classification necessary for both interpretations (i.e., number specifications must reflect the dual character of the noun). This means these nouns will typically receive the NUMBER code “SG/PL”.

change form in the plural, such as “man/men”, and “boy/boys”, but they could also apply to words like “sheep” which can occur in the noun phrases “one X”, “two X”. Any word that can exist in this environment must be given the feature COUNT, but their number must be separately determined for each surface form. These words correspond to the OALD code C and some words with the Brown tags, NN, and NNS.

4.3.2.14 PROPER

This feature is for proper nouns. It corresponds to the Brown tags PPSS, PPS, PN, NP and NPS. This feature applies to the personal pronouns (as if they were a person’s name) as well as third person proper nouns (which are indeed a person’s name).

4.3.2.15 DEG

This feature identifies words, typically adverbs, that can serve as heads of a degree phrase. As discussed in Chapter 3, these include “as”, “so”, “far”, “too”, but can also include some words with the Brown corpus tag QL. Tests for these words are whether or not they can occur in any of the phrases “X many”, “X rich”, “X quickly”, or “X down the road”.

4.3.2.16 NEG

This feature is for negative words, such as “not”. It would also be necessary tag the “n’t” contraction with this feature.

4.3.2.17 UNTENSED

This feature is for infinitival verb forms, and indicates that the verb does not have a PNCODE. It directly corresponds to the Brown tag VB. If the word is of the same form that occurs in the verb phrase "to X to the store" then it is untensed.

4.3.2.18 PASTPART

PASTPART indicates that the lexical entry is the past participle of some verb form, given as the ROOT of the same lexical entry. It directly corresponds to the Brown tag VBN and some of VBD. This is the word form that occurs in X in the following verb phrase:

(18) a. John has/have X to the store.

b. The rock has/have X.

c. Tom has/have X before.

4.3.2.19 PRESPART

This feature indicates that the lexical entry is the present participle of the verb form in ROOT of that lexical entry. It is usually a gerund with an -ing ending and directly corresponds to the Brown tag VBG. The verb form must be able to serve as the X in the following examples:

(19) a. John is X.

b. The rock is X.

c. The dog could be X.

d. The man should have been X.

4.3.2.20 PRED

This feature is given to adjectives which can only occur in predicative position. For example words like "ablaze" which correspond to the OALD code "pred." If the adjective can occur in the sentence "John is X." but can not occur in the sentence "The X man went home." or "The X rock didn't move." then it should receive this feature.

4.3.2.21 LA

Linking + Adjective

This is a verb subcategorization feature that denotes a linking verb that takes an adjective phrase predicate. The best example is the verb "is" in sentences like "The man is sleepy." The sentence pattern underlying this code is [DP V ADJ], where the verb is intransitive and assigns the attribute of ADJ to the DP subject.

4.3.2.22 LN

Linking + Noun (i.e., determiner) phrase

This verb subcategorization feature is just like LA, except the predicate is a noun phrase such as "The man is a doctor." The sentence pattern underlying this code is DP V DP, where the verb is not a transitive verb and the DP is not an object. Rather, the verb serves as a kind of "=" sign to assign a description to the subject.

4.3.2.23 I

Intransitive

This subcategorization is given to intransitive verbs. An example of this is the verb "cry" in the sentence "John cries." No object is required with this verb, although there may be adjuncts, such as in the sentence “John cries profusely” or “John cries in the living room.”

4.3.2.24 IPR

Intransitive + PRepositional phrase

This is a code for intransitive verbs specifying that this verb can occur with a prepositional phrase adjunct. In general, most intransitive verbs can occur with prepositional phrase adjuncts, but this code is included for completeness. An example of a verb with both the I and IPR code is “cry” which is grammatical in either of the two sentences:

(20) a. John cried.

b. John cried in the living room.

4.3.2.25 IP

Intransitive + Particle

This is the same kind of code as described with the previous code; it indicates that a verb can occur with a particle (i.e., “up”, “off”, “away”). Not all intransitive verbs have this code though, exemplified in the ungrammatical sentence “*I like to garden up/away/off”. These particles are not closely associated with the meaning of the verb, for example, both of the following sentences have basically the same meaning:

(21) a.The birds chattered.

b.The birds chattered away.

4.3.2.26 INPR

Intransitive + Noun (i.e., determiner) phrase or PRepositional phrase

Like the previous I codes, this code specifies the kind of adjunct the Intransitive verb can occur with. This code indicates that the verb could take either a determiner phrase or a prepositional phrase and retain the same meaning. An example of this type of verb is “lasted” which is grammatical in either “The meeting lasted for a week” or “The meeting lasted a week.”

4.3.2.27 IT

Intransitive +To- infinitive

This Intransitive verb code is given to verbs that allow a to-infinitive (i.e., an exceptional clause with an unlexicalized subject) to follow them as an adjunct. The verb is still intransitive, so there is no object in sentences with these kinds of structures. An example of this kind of verb is “hesitated” in “John hesitated to phone home.” The exceptional clause can not be an object because “John hesitated” is also a grammatical sentence. Recall from chapter 3 that a verb subcategorizes for its objects, so that if a verb does have an object, it is necessary for completing the sense of the verb. The ungrammatical sentence “*Nathan killed.” is ungrammatical because it does not have the object required for the “killing” action.

4.3.2.28 TN

Transitive verb + Noun (i.e., determiner) phrase

This is the code given to the common [DP V DP] structure where the second DP is the verb object. The verb “hit” has this subcategorization, as revealed in the grammaticality of “John hit the baseball.” The object position is required for this verb because it is transitive, and this code says the structure of that object will be a DP.

4.3.2.29 TNPR

Transitive + Noun (i.e., determiner) phrase + PRepositional phrase

This code is given to Transitive verbs that take a single object made up of a determiner phrase and prepositional phrase (i.e., a small clause of the form [DP PP]). An example of this kind of verb is shown in the sentence “Mary convinced the court of her innocence.” This is in contrast to a sentence like “John saw the movie in the living room” because with “saw”, the prepositional phrase “in the living room” is modifying the action of the verb (i.e., where the “seeing” took place). Conversely, with the “convinced” sentence, the prepositional phrase is more fundamental to the meaning of the sentence. The sentence “Mary convinced the court” is not a complete thought, whereas ““John saw the movie” is.

4.3.2.30 TNP

Transitive + Noun (i.e., determiner) phrase + Particle

This code is for Transitive verbs that take a single object made up of a determiner phrase and a particle (i.e., a small clause of the form [DP ADV]). An example of this is the verb “shook” in “The nurse shook the medicine up.” Like the prepositional phrase in code TNPR, the particle in this code has a close association with the object of the sentence. It is therefore a different construction than would be for a sentence like “I was shaken up.” where the particle is more like part of the meaning of the verb (cf. Frazier, 1991).

4.3.2.31 TF

Transitive + Finite “that” clause (i.e., CP)

This code is for Transitive verbs that take a CP as object. An example of this is the verb “believe” as in the sentence “John believes that the class will be cancelled.” The ungrammaticality of “*John believes[4]” shows that the object is required to complete the meaning of this verb, and this code determines that the structure of that object will be a finite clause.

4.3.2.32 TW

Transitive + Wh-clause

This code is for Transitive verbs that take a wh-clause, or indirect question, as object. There are two forms of complements possible with these verbs, as in the sentences “John decided what we should do next.” and “John decided what to do next.” This is implemented as either a CP object or an EC object introduced by the WH word.

---

4. If this is a grammatical sentence it is because there is an object that is understood as part of the current discourse and so need not be lexicalized. Accounting for this goes beyond the scope of syntax.

4.3.2.33 TT

Transitive + To-infinitive

This code is for transitive verbs that can take an exceptional clause with an unlexicalized subject, so that the underlying pattern is [DP Transitive-verb [PRO to VP]]. An example of this code is in the sentence “Mary hates to drive in the city.” The subject of the exceptional clause is represented by “PRO”” but actually refers back to the subject of the sentence. Transformational grammar refers to this structure as either “Subject-to-Subject Raising” or “Subject-Controlled Equi” because both verbs in the sentence share the same subject.[5]

4.3.2.34 TNT

Transitive + Noun (i.e., determiner) phrase + To-infinitive

This code is like the code TT except here the subject of the exceptional clause is lexicalized. The underlying pattern is [DP Transitive-verb DP [PRO to VP]] where the PRO in this case refers to the object DP. For example, the sentence “John expected Mary to wait for him” could be paraphrased “John expected that Mary would wait for him” where Mary is the subject of “waiting”. The structure this code describes is what Transformational Grammar calls “Subject-to-Object Raising” because the subject of the embedded clause raises to be the object of the main clause.

---

5. Subject-to-Subject Raising and Equi sentences are believed to be derived differently. Here I am not attempting to explain that derivation, only to describe the resulting surface structure. Cf. chapter 5 for further discussion on this issue.

4.3.2.35 TG

Transitive + verb+ inG headed verb phrase

This code describes a structure similar to that for TT; however, the object in this case is a small clause with an unlexicalized subject and a gerundive verb form. The underlying pattern is [DP Transitive verb [TRACE V-ing]]. This is also an example of Subject-dominated sentences, because the subject of the V-ing is also subject of the main clause. The difference is that the verbal inflection is different because there is no INFL in the embedded clause. An example of verbs that take this structure is "enjoys," which produces sentences like "John enjoys playing baseball."

4.3.2.36 TNG

Transitive + Noun (i.e., determiner) phrase + verb + inG verb phrase

This code is similar to TNT in that it is an Subject-Raising structure where the subject of the embedded clauses "raises" to be the object of the main clause. Like TG, this code has a different verb inflection because there is no INFL. This makes the pattern underlying this structure [DP Transitive verb DP [TRACE V + ing phrase]] where the embedded clause is a gerundive small clause. Consider the sentence "The man spotted the children waving from the playground." as an example.

4.3.2.37 TNI

Transitive + Noun (i.e., determiner) phrase + Infinitival verb phrase

This code produces a structure similar to that in TNG except the verb form is infinitival rather than gerundive. The pattern underlying this structure is virtually identical to that in TNG: [DP Transitive verb DP[TRACE V+0]. An example is seen in the sentence "We watched the men unpack the china."

4.3.2.38 CNT

Complex-transitive + Noun (i.e., determiner) phrase + To-infinitive

As I mentioned earlier in this chapter, OALD calls verbs that can take two objects "complex-transitive." What this really means is that the object of the sentence takes its own objects and therefore the sentences is, as transformational grammar describes it, "Object Controlled." To understand the difference between an object controlled sentence like those produced by the code CNT and sentences with apparently the same surface structure but with the code TNT, consider the following:

(22) a. Jane forced the man to give up the money.

b. Jane promised the man to give up the money.

In sentence (22a), the subject of "giving up the money" is "the man", whereas in sentence (22b) it is Jane who must give up the money. Thus, (22a) is the object-controlled, CNT sentence and (22b) is a subject-controlled TNT sentence.

The underlying structure of CNT sentences is identical to that of TNT sentences:

[DP Transitive-verb DP [PRO to VP]].

4.3.2.39 CNN

Complex-transitive +Noun (i.e., determiner) phrase + Noun (i.e., determiner) phrase

This code denotes another object-control structure, but there are no PRO or TRACE in the place of unlexicalized items. The complement structure is a small clause of the form [DP DP] where the first DP is the primary object of the sentence and the second DP modifies the first. An example of this sentence is “The court considered Smith a trustworthy witness.”

4.3.2.40 CNA

Complex-transitive + Noun (i.e., determiner phrase) + Adjective phrase (i.e., degree phrase)

The structure described by this code is identical to that in CNN, except the small clause has the form [DP AP] (where AP is implemented as a DegP). An example of this kind of sentence is “The freezer kept the ice cold.”

4.3.2.41 CNG

Complex-transitive +Noun (i.e., determiner) phrase + Gerundive verb phrase

Another control structure, the form underlying CNG complements is the small clause [DP VP + ing]. This pattern is characterized by the sentence “The policeman got the traffic moving.”

4.3.2.42 CNI

Complex-transitive + Noun (i.e., determiner) phrase + infinitival verb phrase

This object-control structure has complements that are infinitival small clauses of the form [DP V+0]. Consider as an example the sentence "Mother won't let the children play in the road."

4.3.2.43 DNN

Di-transitive + Noun (i.e., determiner) phrase + Noun (i.e., determiner) phrase

Di-transitive verbs take both an object and an indirect object. The object is a variable structure but the form of the object is simply described: it is either a DP or a PP. Verbs with the code DNN have both their direct and indirect objects being determiner phrases. An example is the sentence "John taught the children French" where the first DP is the indirect object and the second is the object.

4.3.2.44 DNPR

Di-transitive + Noun (i.e., determiner) phrase + Prepositional phrase.

This is another form of di-transitive sentences in which the indirect object is a prepositional phrase occurring last and the object is the determiner phrase which immediately follows the verb. This is exemplified with the sentence "John taught French to the children."

4.3.2.45 DNF

Di-transitive + Noun (i.e., determiner) phrase) + Finite clause

For this complement code, the indirect object is a determiner phrase and the object is a finite clause. An example of this is the sentence “The leader told Paul that the job would be difficult.”

4.3.2.46 DPRF

Di-transitive + PRepositional phrase + Finite clause

This code is just like DNF with the exception that the indirect object is a prepositional phrase instead of a determiner phrase. This kind of pattern shows up in sentences like “The President announced to the journalists that he was resuming his duties.”

4.3.2.47 DNW

Di-transitive + Noun (i.e., determiner) phrase + Wh-clause

This code specifies that the indirect object, which occurs directly after the verb, is a determiner phrase and that the object is a Wh-clause. Recall in the description of the code TW that there are two possible variations of WH-clauses. Here they are implemented the same as there, so that the possible sentences are as follows: “The host reminded the guests where to put their luggage” or “The host reminded the guests to put their luggage away.”

4.3.2.48 DPRW

Di-transitive + PRepositional phrase + Wh-clause

The pattern underlying this code is very much like that just described in DNW with the exception that the indirect object is a prepositional phrase rather than a determiner phrase. The indirect object still follows directly after the verb, as in the sentences "You should indicate to the team where they should assemble" or "You should indicate to the team where to assemble".

4.3.2.49 DNT

Di-Transitive + Noun (i.e., determiner) phrase + To-infinitive

The pattern specified here is for a determiner phrase indirect object and a direct object that is a to-infinitive, meaning an exceptional clause with an unlexicalized subject. The underlying structure is just like that in CNT and TNT: [DP Transitive-verb DP [PRO to VP], and indeed some words share more than one of these codes. The DNT sentences are differentiated from the CNT and TNT sentences with regard to the meaning of the verb and the semantic roles it discharges. Consider the sentences in (23):

(23) a. Jane wanted the man to give up the money. [CNT]

b. Jane expected the man to give up the money. [TNT]

c. Jane warned the man to give up the money. [DNT]

Di-transitive sentences can be understood as describing a transferal relationship between the object and direct object. In (23a-b) it is difficult to identify the goal and the transferring action, whereas in (23c) it is clearly a "warning." given to "the man". These verbs can also often be rephrased like "subject gave a indirect object to the object" where the indirect object is specifically described in the object position, but is generally described by the meaning of the verb. For example, in (23c), the indirect object is "to give up the money" and this was the specific warning given to the man.

4.3.2.50 DPRT

Di-transitive + PRepositional phrase + To-infinitive

The DPRT pattern specifies a prepositional phrase indirect object and a to-infinitive direct object. It is a structure similar to DNT in that the indirect object directly follows the verb; however the form of the direct object is a prepositional phrase. Consider the example "Fred signalled to the waiter to bring an extra napkin."

4.3.3 Value features

A value feature is one whose argument is a pre-determined value rather than simply a "t" or "nil" as was the case for toggled features. A typical feature assignment takes the form "<FEATURE> <VALUE>" (i.e.. "NUMBER (SG)"). A chart of features, categories they apply to, and possible values is given below in Table 4.3.

Table 4.3 Value Features and Appropriate Categories

FEATURE	CATEGORY	VALUES
NUMBER	N, PRO, DET, ADJ	SG, PL, SG/PL
PNCODE	V	PRES, PAST, X3SG, 3SG, NONE, ANY
TAKES	DET	SGCT, PLCT, NONCT
ZONE	ADJ	1, 2, 3

Only the values for the TAKES and ZONE are specific to this project; the NUMBER and PNCODE values are macros. The NUMBER values are combinations of the word's person, number, and tense (cf. Chapter 3 on INFL). It would be possible to implement person, number, and tense separately or in a different way, but this would require revising some of the primitives used in the agreement tests associated with this grammar. With this said, I will proceed to discuss the details of the implementation of the numbering scheme used here.

4.3.3.1 NUMBER & PNCODE

The values assigned to both these features are a subset of the variables 1SGPRES, 1PLPRES, 2SGPRES, 2PLPRES, 3SGPRES, 3PLPRES, 1SGPAST, 1PLPAST, 2SGPAST, 2PLPAST, 3SGPAST, and 3PLPAST. The noun's number and person determines the first part of the code, and the verb's tense determines the last part, as described below:

4.3.3.1.1 Nouns

Number itself is strictly a binary feature: singular and plural. Nouns use this feature in conjunction with person to specify verb agreement. Person is a ternary feature having the values first, second, and third. First person is typically only for self-referring personal pronouns like "I","me", or "we"; second person occurs with the personal pronouns "your", "y'all" or "you"; third person is for referring expressions like "John" or "her", but also includes pronouns like "he", "she", and "it" and nouns like "man", "dog", etc. Possible combinations for person and number are:

(23) 1SG, 2SG, 3SG, 1PL, 2PL, 3PL.

The 3SG and 3PL are used most often.

In order to simplify number classification, my implementation uses the macros "SG", "PL", and "SG/PL" to assign number values. Here, "SG" refers to all of 1SG, 2SG, 3SG; "PL" refers to all of 1PL, 2PL, 3PL; and "SG/PL" refers to all six combinations. The tense part of the codes is not relevant to nouns, but when a word has the person and number 3SG, for example, it is given both 3SGPRES and 3SGPAST in order to account for all the verbs that can agree with it.

4.3.3.1.2 Verbs

Noun-verb agreement depends on tense in addition to person and number. This is clearly shown when trying to classify the usage of a verb like "hit", which produces the following grammaticality judgements according to the tense of the verb:

(24)

a. *The man hit John [3SG, Present tense]

b. The man hit John [3SG, Past tense]

c.The men hit John [3PL, Present tense]

d.The men hit John [3PL, Past tense]

The motivation for including the tense in the values for NUMBER is to capture data like this. Using these values, it is easy to specify the number on a surface structure which behaves like "hit" ; the NUMBER assignment in the lexical entry for "hit" is the following:[6]

(25) NUMBER '(3PLPRES, 1SGPAST, 1PLPAST, 2SGPAST, 2PLPAST, 3SGPAST, 3PLPAST)

To make it easier to assign these values to words, my system uses the following macros: "PRES" for all present tense codes; "PAST" for all past tense codes, "X3SG" for all present tense codes except 3SGPRES; "3SG" for 3SGPRES; "ANY" for all codes.

Note that the verb feature PNCODE is only applicable to inflected verb forms; infinitives are given the toggled feature UNTENSED to account for their lacking person and number inflection.

4.3.3.2 TAKES

This feature is assigned to central determiners to specify the kind of noun that a particular determiner can occur with. This is analogous to the verbal subcategorization codes, except it is not a toggled feature. Possible values to this feature are SGCT, PLCT, and NONCT . These are derived from the possible noun types (cf. COUNT, MASS, and PROPER below). These codes should be assigned to determiners according to the paradigm below:

(26) SGCT --> "X man" or "X dog"

PLCT --> "X men" or "X dogs"

NONCT --> "X fish", "X sugar", or "X music" or proper nouns

---

6. Another side-effect of implementing the noun and verb numbers with the same variables is that noun-verb agreement is now a trivial task that can be done with a simple LISP member function.

As is the case with the verb codes, all values which apply to a specific determiner must be included in the argument to the TAKES property. Thus, a typical value assignment for the determiner such as “the”, might be “TAKES (SGCT PLCT NONCT)”.

4.3.3.3 ZONE

This property allows adjectives to be categorized according to the preferred order of their occurrence (cf. Chapter 3). This categorization is based on the modification zones posited by Bache in (Bache, 1978). The first zone is reserved for adjectives which semantically define or specify rather than describe. Examples of this are words like “same”, “usual”, “whole”, and others that denote size, time or age. Syntactically these words never occur in predicate position and can not be compared or intensified with words like “very” or “extremely”:

(27) a. * The very usual steps...

b. * The extremely same smell...

They also can not be coordinated with other adjectives of any zone:

(28) a. * The same and smooth action...

b. *The whole and inexorable web...

A word should be assigned zone 1 if it is ungrammatical in the following structures:

(29) a. *The X and smooth noun.

b. * The extremely/very X noun.

c. *Noun phrase is X.

Zone 2 is given to central adjectives: those Quirk calls the "most adjectival" in (Quirk, 1985). These adjectives can be compared and coordinated like in (27) and (28) and can also occur as predicates. This means if the word is grammatical in the tests given in (29) it is a ZONE 2 adjective. Semantically these words describe or characterize and also include some present and past participles like "exciting", "terrifying", and "exhausted."

My zone 4 is the same as Bache's zone 3, which is for peripheral adjectives that occur closest to the noun in a noun phrase. They are similar to zone 1 adjectives in that they can be expected to fail the tests in (29): no coordination, comparison, intensification, or predication applies. There is one exception: when there is a lexicalized degree head in the degree phrase. In this case it is possible for these adjectives to serve as predicates, as in the following examples:

(30) a. John was too political.

b. The party was more theatrical today.

Semantically, zone 4 words classify the noun and are usually noun or verbal cognates, such as "political", "social", and -able/ -ible adjectives like "washable", and "understandable". Zone 4 also includes nationalities and the specific words "little, "old", and "young".

With the use of these zones, it is possible to explain the data that follows:

(32) a.very rich white American man

b. ?? very white rich American man

c. ??American very rich white man

d. ??rich American very white man

e. ??white very American rich man

f. ??very American white rich man

Bache would explain these data such that the adjective “very” is a zone 1 adjective, “rich” is a zone 2 adjective, and both “white” and “American” are zone 3 (i.e., what I am calling zone 4) adjectives. It is often the case; however, that color words like “white” must occur before the words that Bache describes as being in what I call zone 4. Consider:

(33) a. white American man

b. *American white man

Because of this I have created a separate zone code for color words: they are assigned zone 3. In this way my grammar captures the ordering preference that occurs among adjectives: “very” is zone 1, “rich” is zone 2, “white” is zone 3, and “American” is zone 4.

4.3.4 ROOT

The value of the ROOT property is the uninflected form of a word. For verbs, this means the infinitival form: the ROOT of “hitting” and “hits” is “hit”. The lexical entry for this word would therefore contain an assignment like in the following:

(34) ROOT hit

Comparative and superlative adjective forms are also considered to be inflected forms and will also have a ROOT which is the positive form of the adjective. The lexical entry for the word “greener” would include the property assignment:

(35) ROOT green

Words like nouns and determiners typically do not have the ROOT property in their lexical entry.

Chapter 5

DISCUSSION

In this chapter I will discuss the linguistic strengths and weaknesses of the implementation I have produced. I will go on to discuss the applicability of this project beyond what has already been mentioned. Finally I will identify areas of future work for this project and for attacking the problem of augmentative communication that I motivated in Chapter 1.

5.1 Linguistic Theory

Here I would like to characterize this implementation with respect to Government and Binding syntax, whose formulation of X' theory I have adopted in the grammar described here. I explained in Chapter 3 that I have made use of the functional categories INFL, COMP, DET, and DEG. The existence of these abstract categories is what distinguishes the GB interpretation of X' theory from that of other syntax theories such as Generalized Phrase Structure Grammar and Lexical-Functional Grammar. I chose to adopt these abstract, functional categories because they facilitated applying the X' structure template to all instantiations of X. This allowed subcategorization to be used for explaining not only the relationship between verbs and complements, but also that between determiners and head nouns. Using the DEG functional head enabled capturing the similarities between adjective and adverb phrases. But by far the strongest reason for adopting the DET and DEG functional heads, is because through them I was able to develop a structure to account for the various types of constituents that occur before the head determiner or between the determiner and the noun. The branching structure the functional heads provide allowed me to eliminate ordering tests on the grammar arcs; tests which would have been necessary to ensure well-formed word sequences. For example, without a structural position at the beginning of the determiner phrase (i.e., the specifier position), to account for the sequence "all the many men," I would have needed a looping determiner category arc. Instead I can implement a series of arcs with different category and feature requirements motivated by an overall grammatical theory. This makes the implementation something more than an ad hoc solution to the problem.

It was also important to be able to apply the X' template to any position in a sentence, including the sentence level itself, because this facilitated producing a complete grammar. It was therefore possible to confront a common objection to using grammars in augmentative communication: that complete ones are difficult to construct. A complete grammar is crucial for communication devices because a user will be using the device to produce normal, everyday language. Consequently they must be able to produce all of the syntactic structures that a human language user could think of constructing. The X' template eliminates this difficulty because it gives a standard structure that underlies all syntactic structures. The problem is reduced to providing structures in the grammar and using subcategorization to eliminate those that are inappropriate for particular lexical items.

Borrowing these concepts from GB, the system performs a number of functions in the way GB predicts. It accepts a surface structure and undoes the transformations that show up there so that the structure it produces is akin to the sentence's deep structure. Movement occurs from argument positions and is only allowed to land at appropriate landing sites. Landing sites can be easily determined with this formalism because the process of undoing a transformation must be explicitly invoked (i.e., a hold action is performed in a grammar arc). In this way, the grammar controls for what constituents can move and to where: if a constituent is encountered that is not in an appropriate landing site, then the parser will be unable to complete a parse for that sentence. In this way, Government and Binding theory's constraints on NP and WH-movement are obeyed, even though they are not overtly implemented as such (i.e., there is no instance in the grammar or processing when I invoke some procedure called "NP-movement".)[1]

But even with these GB movement characteristics, a GB motivated X' structure, and adherence to subcategorization in the way that GB's over-arching Projection Principle suggests, the structures this grammar produces are often not those that GB theory would predict.[2]

---

1. This is a consequence of the fact that I am using a rule-based parser rather than a principle-based GB parser such as that described in (Werhli, 1988). The principle-based parser works with a base-generated structure and explicitly applies the GB principles such as Binding, Theta-Criterion, Government, and the Empty Category Principle. In comparison, a rule-based parser focuses on surface structure, and applies pre-determined rules to assign a structure to the input sentence.
2. The Projection Principle, which applies at all levels of syntactic analysis (i.e., deep structure, surface structure, phonetic form, and logical form) was originally given by Chomsky in his Lectures in Government and Binding, 1981. The original formulation is given here, taken from (Sells, 1985):

> Representations at each syntactic level are projected from the lexicon, in that they observe the subcategorization properties of lexical items.

For example, this grammar analyzes a relative clause as the result of a movement of a DP out of an embedded clause and into a higher position in the tree. This analysis is significantly different from the Government and Binding theory analysis, which posits that the position seen in surface structure is the position where the noun phrase originated, or was "base generated."

To represent the movement analysis, the grammar restores the "moved element" to its original position in the sentence while at the same time leaving a copy of that element in the position where it was found.[3] This has the strange side-effect of generating two occurrences of the moved item in the deep structure of the sentence. This action is crucial for the prediction to be effective. Consider, for example, a sentence where the subject of the relative clause is the subject of the main sentence:

(1) The man who walks the dog was late today.

When the predictor has the partial sentence "The man who", if the moved noun phrase "the man" is not put back into the position following "who", the predictor will not be able to eliminate the sentence

(2) *The man who walk the dog was late today.

The predictor will not know what kind of inflection the verb of the relative clause must have.

This problem motivates the necessity of a deep structure with the form:

3. This occurs only with relative clauses and wh-questions. The movement done to analyze passive sentences is simply an exchange of argument positions. The trace DP's are generated only for clauses where there is a "hole" in the surface structure.

(3) The man [who the man walks the dog] was late today.

It is necessary for both occurrences of “the man” to be in the sentence in order for that clause’s subject-verb agreement to be checked.[4]

Conversely, there are some cases of movement where the deep structure produced by my grammar is faithful to the GB analysis. Consider, for example the structure produced for control sentences like the following:

(4) a. John expected Mary to wash the dishes.

b. John expected to wash the dishes.

The GB analysis would predict that the real surface structure of these sentences is like:

(5) a. John expected [Mary to wash the dishes].

b. John expected [PRO to wash the dishes.]

Here, PRO is an empty category that refers back to John. The structure that my grammar will produce is exactly that in (5a-b). The parser neither performed or undid any movement to derive the structure; however, contrary to the GB analysis that “Mary” is the object of “expected” because the NP moves in order to satisfy the Theta Criterion.[5]

---

4. Recall from chapter 2 that in practice, only the noun “man” is replaced into the original position. The head noun of the determiner phrase holds all the agreement information necessary to correctly analyze the sentence and is therefore the only part of the moved constituent that must be maintained.

5. Notice that it is clear that “Mary” serves the object role in the sentence because if the name were to be replaced with the female pronoun, it would be the accusative pronoun “her” rather than a nominative “she”. The Theta Criterion explains that verbs have particular theta roles which must always be discharged. The verb “expect”” requires an object, so it discharges that role by causing the subject of the embedded clause to move into object position in the main clause.

This grammar can therefore be characterized as one that borrows significantly from GB syntax, but is not a complete representation of the Government and Binding Theory of grammar. This stems from the fact that GB is a descriptive theory of grammar. Its definitions of C-command, government, and the empty category principle are theoretical definitions used to describe relationships between words or within syntax trees. The relationships must hold for a sentence to be grammatical: they are a way of describing what has gone wrong in an ungrammatical sentence's derivation.

My purpose, however, is not to describe grammaticality but to implement a grammar such that only grammatical sentences can be produced by the user. The distinction between these two uses of a grammar theory has been discussed by Roger Berwick and Amy Weinberg as the "Type Transparency Hypothesis" (Berwick & Weinberg, 1983). They question to what extent a computational grammar of English should perform sentence parsing the way the theory of grammar predicts; in other words, whether or not the grammar theory is equivalent, or transparent, to the method of parsing. With my grammar and parsing implementation, I have not preserved a transparency between the two components. Rather than explicitly implementing Government and Binding Theory notions like Government, C-command, and the Empty category principle, I have used these principles to guide the construction of the grammar. This means that while the parser does not explicitly check for the relationships these principles denote, they are implicitly at work within it because of the way the grammar has been constructed.

For example, government is a relationship that describes what constituents a head can determine (or influence): it defines the scope of the head.[6] In most cases, government amounts to the sister relationship that holds between the head and its complement. Among others, the concept of subcategorization is said to occur under the relationship of government. This is exactly the case in my implementation: a verb or determiner subcategorizes for only its complement because that is the only position it governs. Therefore I have used the Government and Binding theory to guide the construction of the grammar, but it does not also predict the way the sentences are processed. This explains the occurrence of deviations from GB predicted structures, such as with the analysis of relative clauses I described previously. It also explains how my grammar is licensed to produce the same deep structure for control sentences like those discussed previously, while the method for deriving these structures is different from the theoretical explanation of them.

Berwick and Weinberg have made the same conclusion as that implemented here: that a theory of grammar and a parsing algorithm need not be transparent.[7] A consequence of this is that the grammar theory does not need to be extremely suited for parsing or computational implementation. This became clear in this project because Government and Binding syntax, as a theoretical framework, gives little emphasis to the

---

6. I am assuming the definition given in (Sells, 1985):
α governs β iff
(a) α c-commands β, and
(b) α is an X, i.e., (N, V, P, A, INFL), and
(c) every maximal projection dominating β dominates α.

7. Computational linguists such as Bresnan have not come to the same conclusion and have therefore adopted a theory of grammar that explicitly aims to preserve the transparency between parsing and grammar (Kaplan, et al. 1981). This grammar is called Lexical-Functional Grammar (LFG), having no transformations of the kind posited by Transformational Grammar, but instead depending on complex lexical entries. Robert Berwick describes LFG as claiming to have “all the descriptive merits of transformational grammars, but none of its computational unruliness. (Berwick, 1981)”

details of grammar structures (i.e., what gets attached where). Those structures that are analyzed in detail tend to be only the anomalous or "interesting" ones that test the limits of GB principles. The result is that there is no standard interpretation of X' theory attachments or about the interpretations of particular kinds of complements. For example, it is important for a computational implementation to know where adjectives and adverbs can be attached in the structure or what kinds of constituents can appear in the Specifier positions of all instantiations of X, but these are topics that have received little treatment from the theorists.

Nevertheless, more than any other grammatical theory, Government and Binding syntax was easily adaptable to the requirements of the ATN computational formalism.The formulation of X' theory found in GB exploits the generality of the XP structural template to the fullest.[8] In addition, it allows minimal changes to a generic lexicon of English (i.e., one that includes syntactic categories and little more than perhaps number and agreement information) and this is most often all the information that computational systems have access to. In contrast, grammatical theories like Generalized Phrase Structure Grammar and Lexical-Functional grammar exploit syntactic features and complicated coding systems provided in the lexical entry of each word. Since subcategorization is the only real idiosyncrasy of a GB grammar, it is easy to integrate a GB-based grammar into other already existing systems, such as the flexible abbreviation system discussed in Chapter 1.

8. The versions of X' theory used in Generalized Phrase Structure Grammar and Lexical-Functional Grammar do not make use of functional heads and therefore they can not apply the abstract template to the sentence level or to minor categories.

5.2 Other Applications

In Chapter 1, I described this project in relation to its application in Augmentative Communication. I described its usefulness for improving a flexible abbreviation system and as a syntax module for prediction systems in general. Other uses within Augmentative Communication can be found because the system does not prohibit using statistical information in addition to the syntax it exploits. For example, statistics could be used to rank predicted categories: the next word in the partial sentence “the gold” might have a higher probability of being a noun than a verb.

Significantly, this work also has application outside the field of Augmentative Communication. A speech recognition system addressing issues similar to those I have described here has been developed at Carnegie-Mellon University (CMU) (Hauptmann, et al., 1988)). The ANGEL speech recognition system shares my goal of applying linguistic knowledge to solving problems in language processing: in this case analyzing speech input so that speech can control a machine’s actions. The problem the CMU researchers must overcome is that analyzing speech input is a computationally difficult and costly task. Initial solutions are reminiscent of the flexible abbreviation expansion I have discussed previously. For example, the CMU’s ANGEL speech recognition system tries to solve this task by generating several hundred word candidates for every word actually spoken. Researchers are currently working to efficiently reduce the number of these possibilities by applying linguistic constraints as early as possible. To that end they have developed the MINDS system, a Multi-modal, INteractive Dialog System (Young, et al., 1989), (Hauptmann, et al., 1988).

MINDS tries to use knowledge gained from studies of discourse, especially notions of focus, user goals, and dialog structure to reduce the computer's search space for determining what speech patterns could mean. The MINDS system uses discourse where my project uses syntax, but both systems attempt to predict what the user will talk about next. My project uses prediction to reduce the searching required by a disabled user when trying to identify the word he or she wants to use. Using prediction in this way increases the communication rate the user can achieve while communicating with his or her AAC device. Similarly, the MINDS system is able to improve speech recognition by reducing the searching required by the machine to identify the word it has "heard". This allows the machine's speech processing rate to increase.

In addition, the MINDS project comes from a background similar to the one found in Augmentative Communication. Until MINDS, speech recognition was done with statistics of word frequencies and collocations. These were based on sequences of two or three words, called Markov models. These same Markov models were used in previous AAC prediction models and the speech recognition systems suffered from the same problems found in those systems: the two and three word transition tables give limited success because their look-ahead is too small and so they erroneously eliminate interpretations that turn out to be correct. Also, they are dependent on word frequencies gathered from relatively small amounts of data and so they may not be accurate.

MINDS runs primarily on semantics, or concepts, that its discourse-tracking capability identifies. It combines these concepts with "a set of syntactic networks" to derive possible sentence structures for the concepts. This means the only syntax done is to determine the lexical realizations of the concepts; the syntax is not comparable to that of natural language users. Consider, for example, that within their Navy ship knowledge base, that the frigate “Spark” has been established as being disabled. MINDS predicts the user will ask about the Spark’s capabilities next. The semantic concepts for the dialog exchange are identified as follows: “shipname” is restricted to the value “Spark”, and any “ship-capabilities” concepts.

They then expand these concepts into the syntactic realizations of ways to refer to the Spark- they allow “the ship”, “this ship”, “the ship’s”, “it”, “its”, “Spark” and “Spark’s”. The notion of “ship-capabilities” generates the syntactic realizations of “all capabilities”, “radar”, “sonar”, “Harpoon”, “Phalanx”, etc. They then combine these to generate a highly constrained search space of phrases like “Does it/Spark/this ship/ the ship have Phalanx/Harpoon/radar/sonar?” or “What capabilities/radar/sonar does the ship/ this ship/it/Spark have?”.

This works well in their constrained environment, but in real-world, unconstrained speech recognition, this type of syntactic generation would be impossible as there could easily be an infinite number of lexicalizations. If the system could use a syntactic prediction system like the one I have outlined in conjunction with the discourse and focus information, then the recognition could be increased without depending on a restricted domain.

It is not clear what role syntax plays in the MINDS system because they are mainly concerned with issues at a higher level of language processing (i.e., discourse and focus). Nevertheless, it would seem that when trying to recognize individual spoken words that the system would benefit from some syntactic prediction that could give information about the structure of the partial sentence and use this to predict the category of the next word. This would limit the search space for speech recognition in the same way it does for abbreviation expansion. Given that the motivation for the speech recognition problem is so similar to that of the project I have described here, it is likely that syntactic prediction could be successfully applied to this field of research.

5.3. Future work

Here I have described my work aimed at making augmentative communication devices more efficient and "usable" for the disabled user. This work has focused on how syntax can be used to eliminate the possible expansions of a creative abbreviation entered at run-time. Using a parallel parsing strategy, I have found it possible to reduce the effort required of the user because he or she is offered only the grammatically appropriate words as abbreviation expansions. A pleasant result of this is that the user is less likely to be confused by the words the computer offers as choices since they are always syntactically relevant to the situation. Other ways that the list of possibilities can be reduced and relevancy be maintained could come from applying other kinds of linguistic knowledge of the sort humans use to understand language. For example, discourse tracking is a kind of pragmatics that could be used to give the system knowledge like "since we have been talking about eating breakfast, it is probably the case that "tbl" stands for "table" and not "tablet". Semantics could also be used to reflect the fact that if the user has used the verb "drink" then we expect the following NP to be some inanimate, consumable object rather than a person's name or things like "table". This sort of information would add to the power that syntactic prediction gives the system and eventually the user will have an extremely small and precise set of words to choose from.

More work could also be done at the syntax level, in the form of adding to the kinds of structures the grammar is able to handle. For example, currently the grammar can not handle coordination, ellipsis, or topicalization; all of which are reasonably common in spoken language. The appendix at the end of this work includes a test suite that demonstrates the coverage of the grammar as it stands at this writing. From this it is easy to see where additions to the grammar could be made.

I feel that the present implementation could also be improved through a more critical analysis of the Degree phrase, especially regarding the lexicalizations of the degree head and the relationship of the head to the elements it subcategorizes for. In particular, it would be useful to re-analyze the status of the quantifier phrase and constituents that can occur in specifier position. In (Ernst, 1991), a positional interpretation of these items is given that may allow a more exact specification of word order in the pre-head noun positions of DP's. The task will be to find an explanation for these kinds of phrases that does not sacrifice capturing the generalities between them (cf. Chapter 3)

Finally, in order for this system to be most useful, it must be implemented in conjunction with a large dictionary that includes the subcategorization codes it requires. Suggestions for carrying out this process are mentioned in Chapter 4, but most important will be to automate the process of assigning the subcategorization codes. I have mentioned previously that this is facilitated by working with learner's-type dictionaries like Longman's or Oxford's which exist in computerized form. I have provided the starting point for this by including references to the Brown corpus tags and explicit descriptions of the requirements for assigning a particular code to a word. On this basis, the task of generating a large dictionary for the system should prove easy to overcome.

Chapter 6

CONCLUSION

This thesis represents a successful application of natural language processing to the problem of augmentative communication. A syntactic predictor can be used to speed communication rate because the system draws on the same rules for creating a sentence that the disabled user exercises as he or she forms sentences. Because of this, the computer is able to intelligently anticipate the form of the word the user will type next.

Most instrumental in the success of the syntactic predictor is how well the grammar that it exploits captures the rules in the language. Through adopting a Government and Binding theory of English syntax, I have been successful at creating a grammar that covers a significant number of constructs, including relative clauses, yes-no and wh-questions, passives, and both matrix and embedded sentences with 39 different types of verb complements. The structure provided by the grammar, however, is still not complete: for instance constructs including what I view as the more rare transformations (e.g., ellipsis, topicalization) are not implemented. Allowing the grammar to analyze these structures would not be difficult to implement, however, because X' theory has allowed my grammar to be uncomplicated and therefore amenable to additions. I believe that, with this grammar, I have developed a strong base to which these and other constructions could easily be added. If it will ever be possible to implement a complete production grammar of English, I believe that the first step is necessarily a grammar of the type described in this work. Just as the full prediction system has application to several research problems, the grammar described here can itself be used for analyses of English, modeling human language use in a machine, and most problems of natural language processing and generation. Thus, not only have I devised an enhancement for disabled users' communication, but I have opened the door to enabling a more complete computational model of language.

CITED BIBLIOGRAPHY

Abney, S. (1987). *The English Noun Phrase in its Sentential Aspect.* Ph.D. dissertation, MIT.

Allen, J. (1987). *Natural Language Understanding.* CA: Benjamin/Commings.

American Heritage Dictionary, Revised Second College Edition. (1976). Boston: Houghton Mifflin Company.

Baker, B. R., & Stuart, S. (1985). Communication Mapping for Semantic Compaction Systems. *Proceedings of the 8th Annual Conference on Rehabilitation Technology*, Memphis, TN: RESNA, 122-124.

Bache, C. (1978). *The Order of Premodifying Adjectives in Present-Day English.* Odense University Studies in English. vol. 3.

Bates, M. (1978). The Theory and Practice of Augmented Transition Network Grammars. In L. Bloc (ed.), *Natural Language Communication with Computers.* New York: Springer.

Berwick, R.C. (1981). Computational Complexity and Lexical Functional Grammar. *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics*, Stanford, CA: ACL,7-12.

Berwick, R.C. & Weinberg, A. (1983). The Role of Grammars in Models of Language Use. *Cognition,* vol. 13, 1-61.

Chomsky, N. (1986). *Barriers.* Cambridge, MA: MIT Press.

Cowie, A.P. (1989). *Oxford Advanced Learner's Dictionary of Current English*, Fourth Edition. Oxford: Oxford University Press.

Demasco, P.W., Lillard, M.,& McCoy, K.F. (1989). Word Compansion: Allowing Dynamic Word Abbreviations. *Proceedings of the 12th Annual Conference on Rehabilitation Technology*, New Orleans, LA: RESNA, 282-283.

Ernst, T. (1990). A Phrase Structure Theory for Tertiaries. In S. Rothstein, ed., *Perspectives on Phrase Structure: Heads and Licensing*. Syntax and Semantics 26, New York: Academic Press.

Francis.W. & Kucera, H. (1982). *Frequency Analysis of English Usage: Lexicon and Grammar.* Boston: Houghton Mifflin Company.

Frazier, L. (1991). Parsing Novel Words. Presented at Cognitive Science Colloquium, May 13, 1991. University of Delaware.

Foulds, R.A. (1980). Communication rates for non-speech expression as a function of manual tasks and linguistic constraints. *Proceedings of the International Conference on Rehabilitation Engineering*, Toronto: RESNA, 83-87.

Foulds, R.A., Baletsa, G., Crochetiere, W.J., & Meyer, C. (1976). The Tufts Non-vocal Communication Program. Presented at the Conference on Medical Devices in Rehabilitation. Boston.

Garside, R., Leech, G., & Sampson, G., eds. (1987). *The Computational Analysis of English.* London: Longman.

Griffith, H.W. (1985) *Guide to Symptoms, Illness, and Surgery.* Tucson, AZ: Body Press.

Hauptmann, A.G., Young, S.R., & Ward, W.H. (1988). Using Dialog-Level Knowledge Sources to Improve Speech Recognition. *Proceedings of the 7th National Conference on Artificial Intelligence*, Saint Paul, MN: AAAI, 729-733.

Jackendoff, R. (1977). *X Syntax.* Cambridge, MA: MIT Press.

Kaplan, R. & Bresnan, J. (1981) Lexical-functional Grammar: A Formal System for Grammatical Representation. In Bresnan, ed., *The Mental Representation of Grammatical Relations*. Cambridge, MA: MIT Press.

Keulen, F. (1986): *The Dutch Computer Corpus Pilot Project.* M.A. Thesis, University of Nijmegen.

Marcus, M.P., Santorini, B., & Magerman, D. (1990). *First Steps Toward an Annotated Database of American English.* Department of Computer and Information Science, Technical Report MS-CIS-90-46. Philadelphia, PA, University of Pennsylvania.

McCoy, K.F., Demasco, P., Jones, M., Pennington, C., & Rowe, C. (1990). A Domain Independent Semantic Parser for Compansion. *Proceedings of the 13th Annual Conference on Rehabilitation Technology*, Washington, D.C.: RESNA, 187-188.

Miller, L.J., Demasco, P.W., & Elkins, R.A. (1990). Automatic Data Collection and Analysis in an Augmentative Communication System. *Proceedings of the 13th*

Annual Conference on Rehabilitation Technology, Washington, D.C.: RESNA, 99-100.

Quirk, R., et al. (1985). *A Comprehensive Grammar of the English Language.* London: Longman.

Radford, A. (1988). *Transformational Grammar.* Cambridge: Cambridge University Press.

Rothstein, S. (1991). Heads, Projections, and Category Determination. To Appear in Kathleen Leffel and Denis Buchard (eds.), *Anthology of Phrase Structure Theory* (Tentative title). Dordrecht: Kluwer.

Sells, P. (1985). *Lectures on Contemporary Syntactic Theories.* CSLI Lecture Notes, no. 3.

Stum, G., Demasco, P.W., McCoy, K.F. (1991). Automatic Abbreviation Generation. Forthcoming, RESNA.

Swiffin, A. L., Arnott, J.L., & Newell, A.F. (1987). The use of syntax in a predictive communication aid for the physically handicapped. *Proceedings of the 10th Annual Conference on Rehabilitation Technology,* San Jose, CA: RESNA, 124-126.

Wehrli, E. (1988). Parsing with a GB Grammar. In U. Reyle & C. Rohrer, eds., *Natural Language Parsing and Linguistic Theories.* Dordrecht: Kluwer.

Woods, W.A. (1969). *Augmented Transition Networks for Natural Language Analysis.* Harvard Computation Laboratory Report No. CS-1, Cambridge, MA: Harvard University.

Yang, G., McCoy, K., Demasco, P. (1990). Word Prediction Using a Systemic Tree Adjoining Grammar. *Proceedings of the 13th Annual Conference on Rehabilitation Technology,* Washington, D.C.: RESNA, 185-186.

Young, S.R., Hauptmann, A.G., Ward, W.H., Smith, E.T., Werner, P. (1989). High Level Knowledge Sources in Usable Speech Recognition Systems. *Communications of the ACM,* vol. 32, no. 2, 183-193.

Zagona, K. (1988). *Verb Phrase Syntax.* Dordrecht, Holland: Kluwer.

REFERENCE BIBLIOGRAPHY

Baumgart, D., Johnson, J., & Helmstetter, E. (1990). *Augmentative and Alternative Communication Systems for Persons with Moderate and Severe Disabilities*. Baltimore: Brookes.

Berwick, R. & Weinberg, A. (1983). Syntactic Constraints and Efficient Parsability. *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics,* Cambridge, MA: ACL, 119-122.

Berwick, R. (1985). *The Acquisition of Syntactic Knowledge.* Cambridge, MA: MIT Press.

Borer, H. (1990). V+ing: It Walks like an Adjective, It Talks like an Adjective. *Linguistic Inquiry,* vol. 21, no.1, 95-103.

Bowers, J.S. (1981) *Theory of Grammatical Relations.* Ithaca: Cornell University Press.

Bresnan, J.W. (1979). *Theory of Complementation in English Syntax.* New York: Garland Publishing.

Dowty, D.R., Karttunen, L. & Zwicky, A.M. (1985). *Natural Language Parsing.* Cambridge: Cambridge University Press.

Emonds, J.E. (1985). *A Unified Theory of Syntactic Categories.* Dordrecht, Holland: Foris Publications.

Ernst, T. (1991). *The Phrase Structure of English Negation.* Unpublished manuscript, University of Delaware.

Grimshaw, J. (1982). Subcategorization and Grammatical Relations. *Subjects and Other Subjects: Proceedings of the Harvard Conference on the Representation of Grammatical Relations*. Bloomington: IULC, 35-56.

Hawkins, J. (1990). *A Parsing Theory of Word Order Universals.* Linguistic Inquiry, vol. 21, no. 2, 223-261.

Hoekstra, T., van der Hulst, H., & Moortgat, M., eds. (1981). *Lexical Grammar.* Dordrecht: Foris.

Hornby, A.S. (1975). *Guide to Patterns and Usage in English.* London: Oxford University Press.

Hudson, R. (1984). *Word Grammar.* Oxford: Basil Blackwell.

Jackendoff, R. (1990). On Larson's Treatment of the Double Object Construction. *Linguistic Inquiry,* vol. 21, no. 3, 427-455.

Kimball, J. (1973). Seven Principles of Surface Structure Parsing in Natural Language. *Cognition,* vol. 2, 15-47.

Lasnik, H. & Uriagereka, J. (1988). *A Course in GB Syntax: Lectures on Binding and Empty Categories.* Cambridge, MA: MIT Press.

Li, Y. (1991). X^0 Binding and Verb Incorporation. *Linguistic Inquiry,* vol. 21, no. 3, 399-426.

Mel´cuk, I.A. (1988). *Dependency Syntax.* Albany, NY: SUNY Press.

McCawley, J.D. (1981). The Syntax and Semantics of English Relative Clauses. *Lingua* 53, 99-149.

Musselwhite, C.R. & St. Louis, K.W. (1988). *Communication Programming for Persons with Severe Handicaps.* Boston, MA: College-Hill.

Rothstein, S. (1985). *Syntactic Forms of Predication.* Bloomington: IULC

Sager, Naomi. (1981). *Natural Language Information Processing: A Computer Grammar of English and Its Applications.* London: Addison-Wesley.

Siegel, M. (1980). *Capturing the Adjective.* New York: Garland Publishing.

Speas, M.J. (1990). *Phrase Structure in Natural Language.* Dordrecht, Holland: Kluwer Academic Publishers.

Tennant, H.R., Ross, K.M., Saenz, R.M., Thompson, C.W., & Miller, J.R. (1983). Menu-based Natural Language Understanding. *Proceedings of the 21st Annual Conference of the Association for Computational Linguistics,* Cambridge, MA: ACL, 151-158.

APPENDIX

TEST SUITE FOR X' THEORY GRAMMAR

The following is a computer print out of a test suite meant to illustrate the capability of the grammar currently implemented with the prediction system. The grammar correctly parses all the sentences or phrases included below unless they are marked with a star. The data in this suite are intended to exemplify syntactic structure and therefore, as utterances, they are often semantically ill-formed. Aside from maintaining emphasis on the syntax of the sentences, use of this kind of test data also allows for the small ~600 word dictionary currently implemented with the system to be used for testing.

Simple determiner phrases

the man
the men
a man
*a men

Pronouns

I
you
we
we men
*we man

ZERO DET phrases and Count nouns

men
*man
time
a time
the time
*week

weeks

Proper nouns

Colorado
the English
English

Mass nouns

furniture
*a furniture
the furniture

Alternate and multiple determiners

two countries
*two country
two Mars
all men
*all the man
all the men
*all man
all of the men
twice the money
many men
*many man
all the many men
*all the many man
all the time
*all some time
one man
*one men

Relative Clauses

the uncle that hit the man has chattered
the uncle that hit the man in the forest has chattered
the uncle that hit the man in the forest was hitting the forest
the uncle that hit the man in the forest was hitting
the uncle who the men hit in the forest has chattered
the uncle hitting the men in the forest is white
the uncle the man is hitting in the forest is white
the uncle the men watched hitting has chattered
the uncle whose forest was hitting has chattered
the uncle with who the man chattered is red

Auxiliary (INFL) verbs

the men do not play
the men hit the man

the man would not hit
the man would hit the man
the man would not have been hitting
the man would have been hitting
the man would not have decided
the man would have opened the man
the man would be hitting
the man would not be hitting
the man would not be hitting
the men have been hitting
the men have not been hitting
the men have opened
the men have not opened
the man is hitting
the man is not hitting

*the man would hitting
*the man have would be hitting
*the men have hitting
*the men would opened
*the men not would open
*the men would have not opened
*the men would have been not hitting
*the men been hitting
*the men have been open
*the men have hitting

Verb Complements

the man is white
the man is a man
the man is playing
the men complain about the men
the men chatter away
the men last a week
the men last for a week
the men hesitate to hit the women
the men open the men
the men convince the men of the forest
the men shook the men away
the men believe that the men hit the women
the men decided where to hit the men
the men decided where the men should hit the men
the men hate to hit the women
the men expect the men to hit the women
the men enjoy playing the men
the men spotted the men playing in the forest
the men watch the men hit the women
the men keep the men white
the men consider the men a woman
the men force the men to hit the women

the men got the men playing
the men let the men play in the forest
the men taught the men women
the men taught women to the men
the men told the man that the men hit the women
the men announce to the men that the men hit the women
the men reminded the men where to hit the men
the men reminded the men where the men should hit the men
the men indicate to the men where to hit the men
the men indicate to the men where the men should hit the men
the men warn the men to hate the women
the men signal to the women to hate the men

WH-Questions

who is the man
who is hitting the man
who is the man hitting
what is the man hitting
what is hitting the man
what is the man
which man is hitting
which forest is the man hitting
who has opened the man
who has the man opened
who hit the man
who did the man hit
who chattered

Y-N-Questions

do not the men play
do the men hit the man
would not the man burn
would the man hit the man
would not the man have been hitting
would the man have been hitting
would not the man have opened
would the man have opened the man
would the man be hitting
would not the man be hitting
have the men been hitting
have not the men been hitting
have the men opened
have not the men opened
is the man hitting
is not the man hitting

*has the man hitting the man
*would the man hitting the man
*have the man would be hitting

*would the man have not opened
*been the man hitting
*be the man hitting
*has hit the man been
*did the man hitting
*do the men be hitting

Agreement

the man hits
the men hit
the man walked
the man walks
*the man walk
*boys walks
*boy walks
I work
*I work
you work
*you works
*we works
we works

Passives

the man was hired by the woman
the man was hired
a story was read to the children by the woman
a story was read to the children
*a story was read by the woman to the children
the children were read a story by the woman
the children were prepared by the woman to go to school
the woman prepared the children to go to school

Adverbs

the children went to school slowly
the children slowly went to school
*slowly the children went to school
the story was quickly read to the children
the story was quickly read to the children by the woman
*the woman read the story quickly to the children
the woman quickly read the story to the children

Adjectives

the woman is a good friend
the old woman with long hair is a good friend
the woman with long brown wavy hair is a very good friend
*the woman with long brown wavy hair is a good verb friend
*the woman with the brown long wavy hair is a friend

*the woman with the wavy brown long hair is a friend
the forest is ablaze
*the ablaze forest is burning

Choice questions- coordination

*Did the mother or the father pick up the children from school
*were the children picked up from school or did they ride the bus

Ellipsis

*the children like the clown more than the man does
*the children need more sleep than do the adults

Coordination

*the man is a teacher and the woman is a lawyer
*both the man and the woman are teachers

Conditionals

*if the children are late for school they will miss the test
*if the train is on time then they will meet soon.

"Farewell, o Muses! And you,
Grammar, be gone with them as well-
Lest that damned Syntax of yours
hasten me into my grave!"

Palladius, a Greek poet of Alexandria,
IV-V centuries A.D.