A MODEL FOR THE PERCEPTION OF SYNTACTIC STRUCTURE

James Peter Thorne
Hamish McL. Dewar
Harry Whitfield
Paul Bratley

English Language Research Unit,
University of Edinburgh.

This research is supported by a DSIR Grant

Contract No. ID/102/2/06

A Model for the Perception of Syntactic Structure

I

The native speaker has the ability to understand an infinite number of sentences in his language. It follows from this that he can produce and understand sentences that he has never heard before. The success of the language learner - the foreigner learning a second language or a child learning his first language - can be measured only by the extent to which he has acquired this capacity. A statement of the knowledge which enables the speaker to associate verbal sounds with meanings in an infinite number of cases is called a grammar. A grammar is a theory of the speaker's linguistic competence. Current linguistics divides this theory into three components, syntax, semantics and phonology, and expresses each as a finite and effectively computable set of rules. Of these three components it is syntax that is seen as reflecting the essentially creative aspect of language behaviour. The rules of semantics and phonology are seen as rules for interpreting the infinite number of structures generated by the syntax.

The program described here simulates the process whereby the native speaker applies his knowledge of the syntax of the language to assign structures to sentences in it. That is, it is a syntactic performance model operating from the point of view of the hearer. By definition we are not constructing a model of the hearer. To do this we would have to incorporate procedures for the implementation of semantic and phonological rules. But to construct a model of the hearer we would have to incorporate in it not only semantic and phonological components. We should also have to introduce factors which while not forming a part of linguistic competence itself nevertheless influence its realisation in linguistic performance - memory limitations, shifts of attention, fatigue, for example, to name just a few - phenomena which for the most part are still little understood (1).

Devices which recognise the syntactic structure of sentences when

presented with them in written form have been studied for a number of years now. Perhaps the most striking feature all these devices have in common is that while in many cases they imply important hypotheses about language behaviour none of them represents an attempt actually to simulate it. The devices so far developed fall into one of two main categories. The first is represented by the most advanced of those currently in operation: the Harvard Predictive Analyser. This is essentially a push-down store mechanism and can be shown to be equivalent to a context-free syntax, in this particular case one in which all the rules are [illegible] is [illegible]. That is all the rules are of the form A -> aB , A -> a , where A and B are intermediate symbols and a is a terminal symbol. For an analyser incorporating a syntax of this form to work it is necessary that at some stage every word in the input sentence should be translated into a terminal symbol of the syntax. This is done by a routine which looks up the words in a dictionary where words are listed together with the form class or classes to which they belong and substitutes the associated set of form classes for every word in the input sentence. The fact that in the syntax terminal symbols are associated with - i.e. occur on the right hand side of equations with - intermediate symbols is interpreted in such a way as to make the occurrence of a particular terminal symbol in the input string predict the occurrence of a particular intermediate symbol in the subsequent structure of the sentence. The set of fulfilled predictions comprises an analysis of the sentence (2).

The notion of prediction employed here is a very natural one and the fact that listening to a sentence we can at almost any point make a fairly accurate prediction about at least part of its subsequent structure is probably an important factor in the efficiency with which we ourselves process sentences. On the other hand it seems unlikely that in every case the first stage in our processing a word is analogous to looking it up in a form class dictionary. Another shortcoming of the Harvard Analyser viewed as a model for human behaviour - which in all fairness it must be emphasised

that it was never intended to be - is that unambiguous self-contained constituents of a sentence are analysed each time they are encountered on a particular analytic path through the sentence. This means that they may be analysed - and each time assigned the same analysis - several times in the course of the processing of the sentence. The efficiency of the hearer makes it unlikely that this kind of repetition occurs in his analysis of the sentence. A third weakness of the Harvard Analyser in its present form applies equally whether it is considered simply as an automatic parser or as a model for linguistic performance. The grammar upon which it is based is, as was pointed out above, equivalent to a context-free phrase structure syntax. But a syntax of this kind is demonstrably incapable of providing adequate descriptions for certain kinds of sentences. Steps are being taken, however, to remedy this deficiency. In a revised model the analysed string produced by the device described will become the input to another component which will incorporate information about transformational rules.

The second type of automatic parsing device currently under development employs the approach of analysis by synthesis. Here more direct use is made of a transformational syntax. On the basis of the number of words in the input sentence the set of all the rules that could have possibly entered into its generation is constructed. Each possible combination is then tested in an attempt to re-generate the original sentence. Each successful attempt represents an analysis of the sentence. Analysis by synthesis reflects very clearly the fact that knowledge of the structure of a sentence is equivalent to the knowledge of how to produce that sentence. And it may well be that recognition of the structure of a sentence involves the hearer in some sense re-generating it. Its weakness as a model for the perception of syntactic structure (and again it must be emphasised that no such claim has ever been entered on its behalf) is that in order to limit its operation to a reasonable time span it too has to operate not upon sentences as such but upon strings of symbols produced by looking up the words in a form class dictionary (3).

It would seem reasonable to suppose that a model for syntactic perception should differ from an automatic parsing device - as these are now conceived - at least in not having as an essential first stage a routine that looks up every word in a form class dictionary, and in not employing procedures which involve an unambiguous constituent being analysed more than once. It would also seem to be mandatory that it should be based upon a transformational grammar and that, therefore, its output should be a statement of the deep structure and not merely the surface structure of a sentence. Since in many cases the deep structure of a sentence is considerably distorted by the transformational rules that produce the surface structure from it, it is likely that this will mean that the model must consist of two components - a deep structure-analyser and a surface structure-analyser. It is also likely that some use will have to be made of the notion of prediction.

The reason for rejecting a dictionary look-up routine that operates upon every word in the sentence can be partly explained by pointing out that in most cases the information obtained in this way increases the complexity of the analytic process rather than simplifies it. For instance, it is the case that in English virtually every word that can be a noun can also be a verb. Each such word would have to be entered in the dictionary as being both parts of speech. The first stage in the analysis of a sentence in which such a word occurs would be to label it as being either a noun or a verb. Now a decision has to be made as to which of these two roles it is playing in the sentence. This is done by reference to the unambiguous words in the sentence. It would therefore seem more economical - if nothing else - not to derive information about such a word from a dictionary but simply to work out the part of speech it belongs to in any particular case by analysis - that is in terms of its fulfilling predictions based upon the occurrence of non-ambiguous items in the same sentence.

This is not to suggest that a dictionary look-up routine can be dispensed with entirely. But there are clearly advantages in restricting

the words contained in the dictionary to those which are syntactically unambiguous. However, in the case of the model described here the mere fact of being unambiguous is not taken as a sufficient reason for including a word in the dictionary. Consider the two words adore and the. Both are syntactically unambiguous. Now consider the sentences He experienced a great adore and He experienced a great the. Clearly the latter sentence represents a far more radical departure from the rules of normal English structure than the former. This suggests that whereas it is possible that a use might be found for adore as a noun it is hardly likely that a new syntactic rule could be found for the. This is obviously connected with the fact that the class of words to which the belongs (determiners) is finite or 'closed', whereas the class to which adore belongs (transitive verbs) and the class to which it was assigned in the nonce sentence given above (noun) are infinite or 'open'. It is this fact - that a word belongs to a closed class rather than its being syntactically unambiguous - that decides its inclusion in the dictionary, although it is a point of some significance that many of these classes are made up almost entirely of words which belong to no other class. Examples of such classes include those sometimes called the 'grammatical formatives'; the determiners, pronouns, auxiliary verbs etc. Examples of closed classes which contain ambiguous words are double object verbs and verbs taking a sentential complement. Though one cannot be as confident in the case of these classes as one can in the case of the grammatical formatives, it seems fairly safe to assume that the chances of new words entering them are fairly slight.

The reason for restricting the dictionary to closed class items is that a considerable reduction can thereby be made in the complexity of the analysing procedures. The dictionary look-up routine reduces the input sentence to a string of symbols indicating closed-class items and the symbol indicating only that a word belongs to one of the open classes. (In each case note is taken of the possible presence of inflexions.) Predictions of the structure of the sentence are then made almost entirely in terms of the

closed-class items - only very general predictions being made on the basis of the occurrence of open-class items. Notice that the analyser does not work with a fully explicit statement of the rules of the syntax but with a greatly simplified generalisation of these rules produced by restating them in terms of open- and closed-class items.

Behind this approach lies the suggestion that in recognising the structure of a sentence it may not be necessary for the hearer to have to be able to work all of the time with all of the rules of the syntax. To make the whole of the syntax available to the program at this stage is to provide it with too much information. Too much in the sense that it is more than is needed, too much in the sense that it is more than can be conveniently handled. It is not necessary to describe the surface structure of a sentence fully in order to reconstruct its deep structure - on the other hand once a description of the deep structure has been obtained this can be used to provide a complete description of the surface structure. The algorithm described below, using only a limited amount of information and processing each item in the sentence only once, decides how many deep structures (kernel sentences) there are in a sentence and what are their surface structure correlates. For example, at the end of this stage - the surface structure analysis - the sentence The boy standing on the corner laughed will have been divided into The boy laughed and ... standing on the corner. Moreover in the course of making these decisions The boy will have tentatively been marked as a noun phrase subject and the deletion of the subject in the second element will have been detected, and laughed and standing on the corner will have tentatively been marked as verb phrase predicates. This output is the input to the deep-structure analyser among whose tasks is reconstructing the deleted subject of the constituent sentence and fitting the constituent sentence into its right position in the matrix sentence.

II

The action of the algorithm may perhaps most easily be described by giving a couple of very simple examples. It must be borne in mind that those examples are intended purely as illustration, so that a great deal of the detail has been suppressed. At almost every stage, for instance, the algorithm we are currently using takes into account more possible predictions and assignments than are actually shown in the examples.

The algorithm works by assigning to every word in a sentence a mark, which indicates in some sense the position of that word in the syntactic structure of the sentence. The assignment of a mark is the fulfilment of a prediction. Thus, suppose we have available the following marks:

(denoting the start of a subject

) denoting the start of a verb (and hence the end of a subject)

> denoting the start of an object

\+ denoting the continuance of a noun phrase.

Then we may mark the sentence The girl likes the sailor as:

(The +girl)likes >the +sailor

Each word has been assigned a mark, and the marking indicates the surface structure of the sentence, namely, one subject, one verb, and one object.

Two minor details may be mentioned here. First, no account is taken of such orthographic features as capital letters at the start of sentences, or of abbreviations ended by a full stop. Secondly, certain elements of punctuation may be treated as words, in the sense that they may create new predictions, terminate certain current analyses, or even be assigned a mark. It turns out, for instance, to produce a pleasant symmetry in the algorithm if we have a mark, say

XX denoting the end of a sentence

which may be assigned only to some such punctuation symbol as a full stop.

Thus, referring to the example above, the analysis would yield in fact

(the + girl) likes > the + sailor XX .

where we note now i) the absence of an initial capital letter
and ii) the assignment of a mark to the full stop.

Let us now follow through the analysis of this sentence.

The progress of the analysis is controlled by a prediction tree and recorded by an analysis tree. We shall speak of the top level of both trees, meaning that level nearest the top of the page in diagrams, even though, as we shall see, the analysis tree 'grows' upwards while the prediction tree 'grows' downwards. Initially, the analysis tree is empty, but the prediction tree contains the two symbols XX (end of sentence) and ((start of subject).

<- immediate predictions

analysis predictions (0)

The top level of the prediction tree contains the immediate predictions, which are marked on the diagrams by a small arrow. The sentence is now read in one word at a time, and the two trees are modified according to the words encountered.

In the sentence we are considering, the first word read in is the. The immediate predictions are surveyed, to see which of the marks they predict may be hung on to this word. The only prediction on the tree is that of (, and this may, by the rules of the algorithm, be applied to the, so this assignment is made. The assignment of a mark results in a modification to the analysis tree, the mark assigned being added on at the current level to those analyses which led up to the prediction just fulfilled. (As we shall see later, more than one possible analysis may result in the same immediate prediction.) In the present instance, only one analysis led to our prediction, and the analysis tree becomes

the

analysis (1)

The assignment of a mark also alters the prediction tree, the par-

ticular alteration depending on the particular mark. When (is assigned, the effect is i) to remove it from the prediction tree
and ii) to put in its place)

This corresponds to the idea that (roughly speaking) every subject has to have its verb with it in the sentence.

One further modification to the prediction tree is also necessary: the word the cannot be a noun phrase by itself, so that it is imperative that there be at least one continuation word. We show this by making the add to the prediction tree, as an immediate prediction, the mark + . The prediction tree therefore is now

<- immediate predictions

predictions (1)

The next word encountered is girl. The search through the closed-class dictionary will not find this word, so all that is known about it is that it is an open-class item. The only immediate prediction is + , and as the rules say this mark may be assigned to an open-class item, the mark is put on the analysis tree, which becomes

girl
the

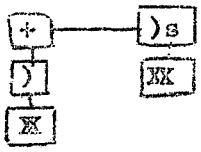
analysis (2)

The effect on the prediction tree of assigning the mark + is
i) to leave the present tree, unchanged, as one immediate prediction
and ii) to remove the mark + from the tree, and promote a copy of whatever lay beneath it as a second immediate prediction.

This corresponds to the idea that, in the absence of further information, a 'continuation' item may be the last element in its noun phrase, or, on the other hand, that there may be more 'continuation' items to come. We notice, however, that in case ii) above the word to which the + has been assigned, namely girl, is an unmarked open-class item, and the last item of

its noun phrase, so that the noun phrase is singular if this analysis is correct. On promoting the) , therefore, it is given a suffix to indicate that any verb to which it is assigned must also be singular.

Thus the prediction tree has now become



<- immediate predictions

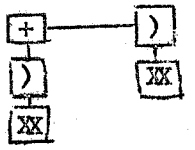
predictions (2)

When the next word, likes, is looked up in the closed-class dictionary, it will be found that it is an open-class item, but that, unlike the previous word girl, it is marked with an important ending, namely -s. Very roughly, this ending is the sign of a plural noun or a singular verb, and also, a fact of some importance for the algorithm, it marks the end of a noun phrase. We find, for instance, the magazine rack, but not *the magazines rack.

(At the risk of stressing the obvious, it would be as well to make explicit some of the more necessary restraints on the generality of the last paragraph. For our purposes, words which end in, say, -ous or -ss do not end in -s: thus, while likes may be a plural noun, dangerous or grass may not. Similarly, words ending with -s' or -'s are in a different class entirely. The closed-class dictionary is expected to take care of distinctions as trivial as this. Any real oddities, such as lens or news, which may both be singular and which, furthermore, do not necessarily terminate a noun phrase - the lens holder, the news bulletin, for instance - are simply listed in the closed-class dictionary with their peculiarities.)

The immediate predictions are two in number on this occasion (2). Consider first the prediction of a continuation, + . This mark may perfectly legitimately be applied to any open-class item, with the effects on the analysis and prediction trees described above. Considering the prediction tree only, for the moment, and considering moreover only the branch with + as an immediate prediction, we see that the assignment of the

mark would transform it to



<-

predictions (2a)

as before.

This time, however, the open-class item to which the + was assigned was marked by a terminal -s. This means that

i) as remarked above, this item must terminate its noun phrase if it has + assigned to it, and therefore any immediate prediction of yet another continuation mark can be immediately declared incorrect;

ii) any noun phrase that this item terminates is plural, so that a promoted verb can be marked accordingly.

The intermediate form (2a) of this part of the prediction tree therefore is transformed still further, one incorrect branch being removed entirely, so that it becomes simply

predictions (2b)

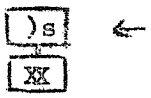
Now consider the alternative assignment to this word <u>likes</u>, that of)s. The rules allow this mark to be assigned to an open-class item provided that this item is marked by a terminal -s or -ed, and it is clear that the former condition is fulfilled. This alternative analysis is therefore entered on the analysis tree, and the prediction tree transformed appropriately. Let us assume (perhaps the wildest over-simplification yet) that a verb may have no objects, one object, or two objects, ignoring for clarity's sake all the complications of verbs which take one complement, an object and a complement, or whatever. (For the purposes of illustration, use is not made of the fact that double-object verbs will be listed in the closed-class dictionary.) Then assigning a verb-sign,) , has the following effect on the prediction tree:

i) the verb-sign's successors can be promoted to be immediate predictions (with, of course, their successors coming up with them);

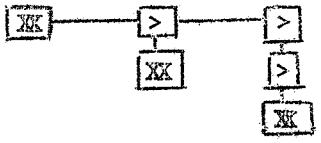
ii) an immediate prediction of one object, followed by predictions of the verb-sign's successors, can be made ;

or iii) an immediate prediction of one object, followed by another object, followed then by the verb-sign's successors, can be made.

Thus, to illustrate, the right hand branch of (2) would be transformed by assigning the)s from



into

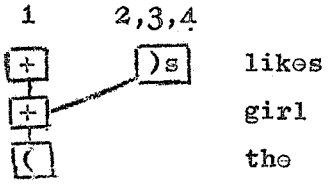


(2) (2c)

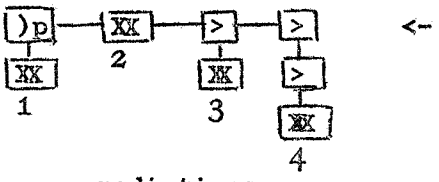
right hand branch

To repeat the same thing in a different idiom, we may visualise any immediate prediction as the top of a pushdown list of predictions. The effect of assigning) is to pop it off the top of the list, and then to continue predicting with the three lists produced by pushing down no, one, or two object predictions onto the result.

The complete picture after all possible assignments to likes have been made, therefore, shows that we have two current analyses and four immediate predictions.



analysis (2d)



predictions

We notice two things. First, we must keep track of which analysis led to which prediction chains. This is shown on the diagram by the numbers above the analysis tree and below the prediction tree. The numbering and the correspondence must be updated whenever an assignment is made, if at the end of the sentence we are to be able to unravel its

analyses, and not merely to answer the question 'Is this sentence grammatical or not?'.

Secondly, we notice that two immediate predictions are the same. When this happens, the similar predictions are conflated so that the next word need only be tested once to see if this particular mark may be applied. In other words, although an ambiguity in an earlier part of the sentence may have given rise to two or more analyses, if the predictions become the same again at some stage, then the analysis of the rest of the sentence is only done once. We feel that this models human behaviour fairly well. Suppose, for example, we come across a sentence beginning:

If the girl guides fish in my river, then ...

Now it seems highly unlikely that the analysis of whatever follows then should be influenced by which of the two (or perhaps more) possible syntactic analyses is chosen for the first clause. Whether little Eva is going to steer her pet goldfish, or whether 16 Troop are going poaching, becomes irrelevant for syntactic purposes once the comma comes along. This is modelled in the algorithm by the fact that the predictions would become the same at the comma for both possible previous analyses, so that they could be conflated into one prediction, which could then be carried on and matched to the rest of the sentence.

It should go without saying that, in the full algorithm, mere identity of predicted mark is not a sufficient condition for the conflation of two predictions. However, in the extremely simple case we are considering, this will be sufficient, and we may conflate the two immediate object predictions of (2d). The subordinate predictions of these two immediate predictions would also be conflated if this were possible, but it is not, so that we are left with alternative predictions one level down, instead of at the immediate level.

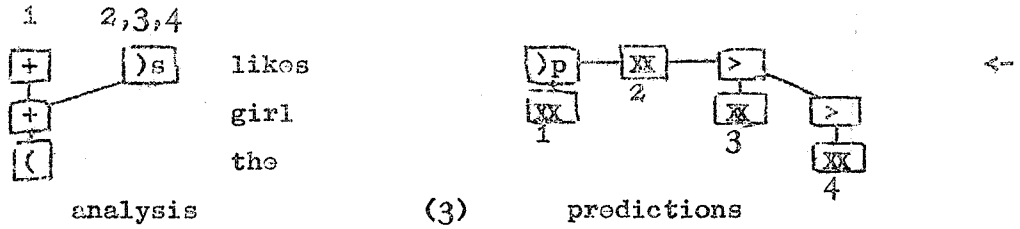
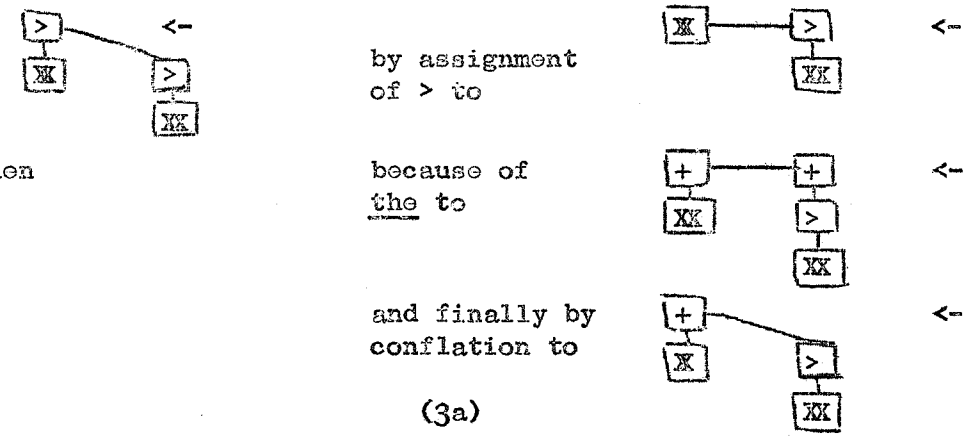


Figure (3) therefore shows the state of the analysis and prediction trees immediately after the word likes has been processed.

The second the of the sentence is now encountered, and each of the immediate predictions is checked in turn to see if it predicts marks assignable to this word. It is clear that the first two immediate predictions fail, as the definite article is not a plural verb and therefore it cannot take)p, and similarly it cannot take the mark XX, which is assignable only to a full stop.

The third immediate prediction, that of an object, is, however, more successful, as the rules allow > to be assigned to the. The effect of assigning this object mark is simply to remove it from the tree, and to promote its successor or successors to be immediate predictions. As we noticed above, however, the word the puts a continuation symbol, + , on the prediction tree. It is clear that this symbol would need to be added in front of all the newly-promoted immediate predictions, and then, of course, by the principles outlined already, all these predictions of + would be conflated. In other words, the right hand branch of (3) may be visualised as being transformed through the following stages:



(3a)

In fact, of course, we proceed straight from the initial to the final state.

Whenever an immediate prediction cannot be fulfilled (as, for example, when)p cannot be assigned to the), then the effect on the prediction tree is simply that this immediate prediction and all its successors, if it has any, are removed from the tree. As no mark is assigned when a prediction fails, nothing is added to the analysis tree.

In the case we are considering, when > is assigned to the, this is a fulfilment of predictions 3 and 4 (see the numbering of fig. (3)), and therefore, as we have mentioned earlier, this mark is added at the current level of the analysis tree to those analyses (in this case there is in fact only one) which led to these predictions. Prediction 1 failed, so no mark is added to that branch of the analysis tree which led to this prediction.

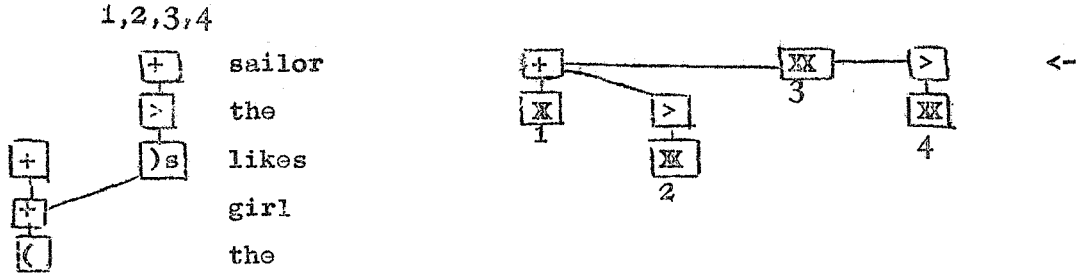
The new predictions and analysis tree are now renumbered, to enable us to keep track of the various analyses, and the result is that after the second the of our sentence the analysis and prediction trees look like this:

analysis (4) predictions

The last word of the sentence, sailor, is now read in, and it is discovered after searching the closed-class dictionary that the word does not appear there, and is therefore open-class and unmarked. To such a word the symbol + may be assigned, and therefore the sole immediate prediction can be fulfilled.

Assigning the mark + has the effect on the prediction tree already described, namely that the present branch of the tree is left unchanged as one immediate prediction, and that the successors of the mark + are promoted, in this case, as there are two immediate successors, to form two new immediate predictions. There is no verb mark,) , among the successors,

so we have no need to do any suffixing. The assigned mark is added to the analysis tree in the usual way, so that the two trees are now:

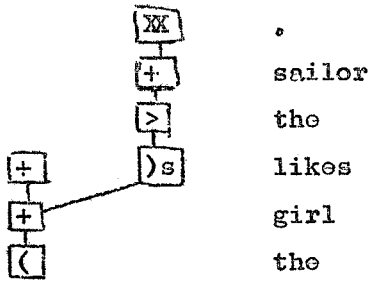


analysis (5) predictions

Finally the full stop is read, which means that the end of the sentence has been reached. The only mark that may be assigned to a full stop is the end-of-sentence mark ⊠, and therefore predictions of any other mark fail. The assignment of ⊠ is represented in the usual way, by adding this mark to the appropriate branch or branches of the analysis tree.

The effect on the prediction tree of assigning this mark is simply to remove it from the tree (no predictions are made beyond the end of the current sentence), and, as all predictions which fail are also removed from the tree, it is clear that when a full stop has been encountered and dealt with the prediction tree will be empty.

The analysis tree, on the other hand, will now be complete, and any path through the tree which starts at the bottom and extends up to finish with the end-of-sentence mark ⊠ will represent a possible analysis of the sentence. In the example we have been considering, for instance, the analysis tree ends up looking like this:



(6)

and it is clear that the only possible analysis of the given sentence is

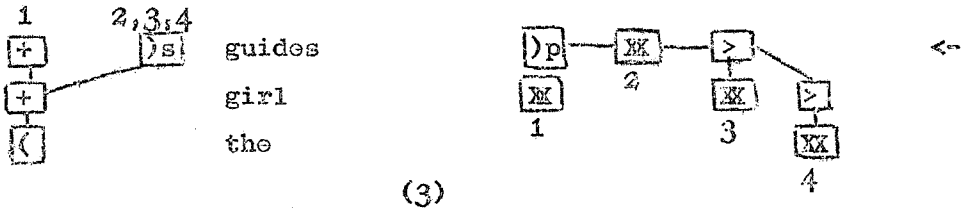
(the + girl)s likes > the + sailor X .

Branches of the analysis tree which do not extend all the way up to the top level represent tentative partial analyses which had later to be abandoned. At present these are retained on the tree in order to be able to follow more easily the working of the algorithm, but there is clearly no strong reason for retaining them if working space is short.

A second brief example may help to make clear what happens to the analysis tree when we analyse an ambiguous sentence (syntactically ambiguous, anyway). Consider the sentence

the girl guides fish.

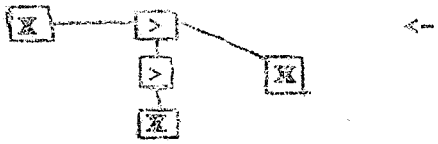
It is clear that the analysis of this sentence will proceed exactly as the analysis of our previous example, up to and including the word guides (which, as far as we are concerned, is exactly the same as likes, being simply an open-class item marked with -s). Thus, when the first three words have been dealt with, the analysis and prediction trees will be, as before,



(3)

This time, however, when the word fish is encountered (an unmarked open-class item), we find that two of our predicted marks may be applied, namely)p and > .

If)p is assigned, then this mark is added to the left hand branch of the analysis tree, as it was this branch which led to this prediction, as we can tell from the numbering of branches and prediction chains. We may again simplify, as we did before, and say that a verb may have no, one, or two objects, so that the left hand branch of the prediction tree is transformed, after one conflation, to



(7a)

XX cannot be assigned to fish, so the middle prediction fails. The symbol > may be assigned, however, in which case this mark is added to the right hand branch of the analysis tree (that branch which led to the prediction >). The assignment of > to an unmarked open-class item alters the prediction tree by

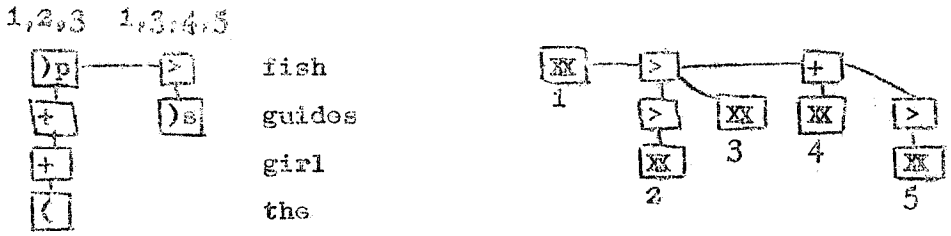
i) promoting the successors of the > to be immediate predictions ; and also ii) making an immediate prediction of + , with the same successors as the > used to have.

This corresponds to the idea that an unmarked open-class item may be either a complete object, or simply the first word of a noun phrase which is the object. (This complication did not arise in the previous example as the insists on at least one successor.) Thus the right hand branch of (3) becomes

(7b)

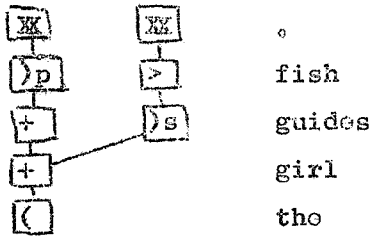
The new prediction tree is formed by joining (7a) and (7b) and then conflating. Notice that as the chain

is common to both parts, we may conflate, and represent it only once. This time we see that the same prediction chain number may occur more than once on the analysis tree since, evidently, more than one method of analysis has resulted in the same prediction chain (after conflation of the prediction tree). The two trees are now



(8)

Once again, encountering the full stop makes all predictions fail except that numbered 1. We see, however, that this time the mark XX was predicted from <u>both</u> branches of the analysis tree, and that therefore the assignment of this mark must be represented by adding it on to <u>both</u> branches. As always after a full stop the prediction tree is empty; and at the end of the sentence the analysis tree has become



(9)

Notice that no attempt is made to conflate the top of the analysis tree, as this would make the separate analyses impossible to untangle.

Now as any path from the bottom of the analysis tree up to the end-of-sentence mark X represents a possible analysis of the sentence, we see clearly that two analyses have been produced for our present example, in other words, that the sentence is syntactically ambiguous. The two analyses are

(the + girl + guides)p fish XX .

and (the + girl)s guides > fish XX .

It will be apparent that the algorithm produces simultaneously all possible analyses of any ambiguous sentence.

It is clearly essential that the algorithm should be able to deal

not only with such simple sentences as those we have so far considered, but also with sentences containing more than one deep structure. However, it is not the task of this part of the total model to do more than analyse the surface structure of the incoming sentence. After this initial pass, a second stage deep-structure analyser will be required to disentangle the surface structure, now signalled by the marks assigned in the first stage, to discover the transformation or transformations used in the construction of the sentence.

Once again, to show how the algorithm works in the case of a sentence with several deep structures, it will be simplest to work through one or two examples.

Consider first the sentence The sailor who kisses her is handsome. This we would mark as:

(the + sailor [who] kisses > her) is = handsome XX .

Several new marks have been used in this example, and we start by explaining them.

[, like (, denotes the start of a subject, with the difference that [brackets the subject of an embedded clause. The meaning of this is almost self-evident, an embedded clause being in some sense buried inside another clause or phrase, as who kisses her is buried inside the subject the sailor who kisses her in the above example.

] is the closing bracket corresponding to [, as) corresponds to (

= denotes the start of a complement.

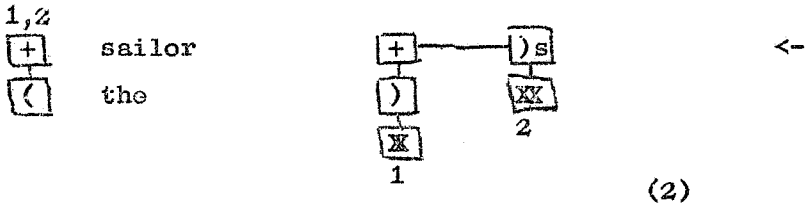
As before, the terminology we are using for this simple demonstration of the algorithm is being applied, and should be interpreted, fairly loosely. The only exact statement of the algorithm is the program defining it, and any other account, particularly a highly simplified one such as this, must of necessity leave much to be desired where rigour and exactitude are concerned.

Another mark which we shall shortly require is

ø the null mark.

This mark is applied to certain words, for example prepositions and conjunctions, which have not been predicted but which themselves create predictions. For example, if we have a sentence which begins In the morning ,. there is little to say about the position of in in the sentence structure except that it happens to occur where it does ; this word would accordingly be assigned the null mark. This is not to say, of course, that no action is taken on discovering this word. On the contrary, the finding of a preposition would cause the prediction tree to be modified in an appropriate way, so that the analysis of the rest of the sentence would, as it clearly must, proceed quite differently from the way it would proceed if a preposition had not been found.

Returning to our example, it will be clear that analysis of the first two words proceeds exactly as in the previous cases discussed in detail, so that when the sailor has been read, the analysis and prediction trees are:



(2)

(Remember that sailor and girl are exactly equivalent as far as our algorithm is concerned: both are simply unmarked open-class items.)

The word who is now encountered, and the effect of this on the prediction tree is as follows:

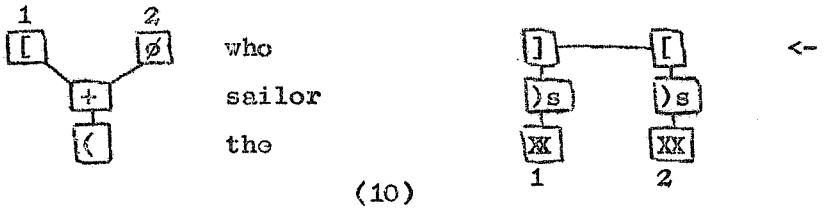
i) any prediction of a continuation sign, + , is made to fail (that is, the part of the tree starting with this prediction is removed)

ii) the mark [is assigned to who, and the mark] made an immediate prediction, its successors being all that is left of the prediction tree after the deletions specified in i)

iii) the null mark ø is assigned to who, and the mark [made an immediate prediction, with the remains of the old tree for its successors just as in ii) above.

These modifications to the prediction tree are made in accordance with the following notions. First, no noun phrase can have a relative clause embedded in the middle of it (*the truck which was a big one driver ..) so that when a relative pronoun is found any predictions of a continuation of the noun phrase must fail. Secondly, there is a type of relative clause in which the relative pronoun is the subject of a following verb, as, for example, the girl who hit the boy, and in this case we want our subject brackets, [ ] , to be round the pronoun; there is also, however, a type of relative clause in which the relative pronoun is an object of a following verb, as the girl who the boy hit, and in this case it must be predicted that the subject will come along later. To these cases correspond modifications ii) and iii) to the prediction tree.

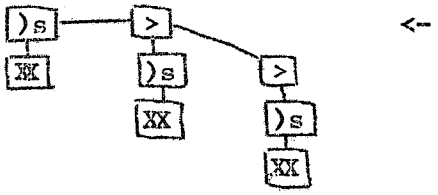
Thus, in the example we are considering, after who has been dealt with, the two trees look like this:



(10)

The next word read is kisses, which would be found to be an open-class item marked with -s, in other words, as we have seen before, a plural noun or a singular verb. Either of our two immediately predicted marks may therefore be assigned to this word.

Assigning] has exactly the same effect on the prediction tree as assigning) , that is, it predicts possibly no objects, possibly one object, or possibly two objects in exactly the same way. The left hand branch of the prediction tree, therefore, after the usual conflation, becomes



(10a)

On the other hand, assigning [to a word is not quite the same as assigning (. The effect of this assignment is, in fact, to remove [from the tree and to put in its place], just as the effect of assigning (was to remove it and to put in its place) ; but on this occasion we must also add a suffix, r, to the verbal bracket] . The reason for this is evident if we look back a couple of paragraphs to where we discussed the effects of the word who. It was explained there that the assignment of ø to who and the prediction of [were the results of predicting the relative clause to be the kind in which the relative pronoun is an object of a following verb. It is this following verb to which we are going to assign the verbal bracket] , and the suffix r is added to the bracket to indicate that the verb has had, in effect, one of its objects already. This now means that when]r is assigned, the possibility of there being two further objects for this verb need not be predicted; there may be no further object, or one further object, but if there were two, this would mean that, counting the relative pronoun as well, the verb would have had in all three objects, which (on our simplified assumptions) is not possible.

We notice also on this occasion that [is assigned to a word marked with -s, so that

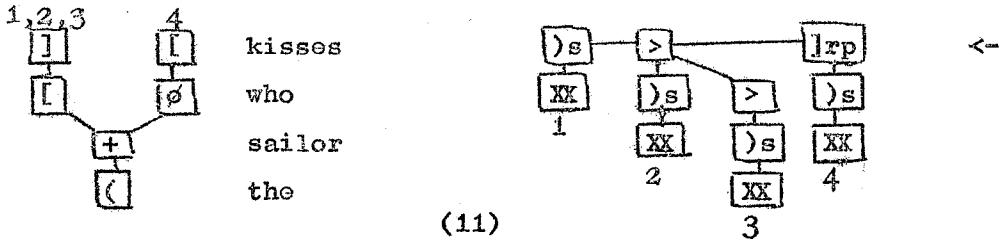
i) there can be no continuation to this noun phrase, that is, no immediate prediction of + is necessary

and ii) the verb of which this word kisses is to be the subject can be immediately marked as plural.

The right hand branch of the prediction tree (10) therefore becomes

<- (10b)

The complete analysis and prediction trees are now:



(11)

The analysis of the rest of the example follows very swiftly, as both her and is will be found on looking through the closed-class dictionary to be words which can only play a very limited set of roles in the syntactic structure of a sentence.

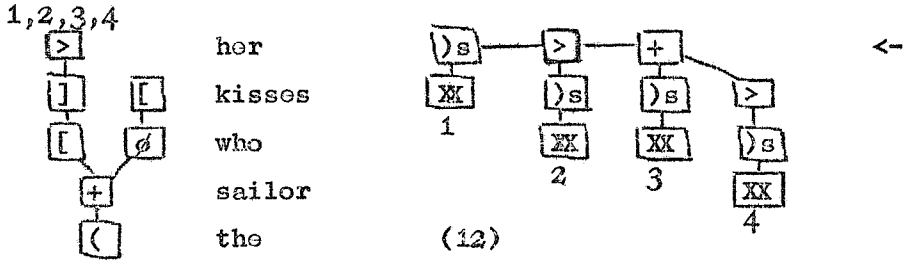
When her is read, the immediate predictions are)s, >, and]rp, and it is immediately apparent that only the middle prediction succeeds, as her is certainly not a verb. Moreover, when the object mark > is assigned to her, it is necessary to make the same predictions as if > had been assigned to an unmarked open-class item, namely

i) to promote the successors of the > to be immediate predictions

ii) to replace the > by +

Case i) deals with the instances where her is an object pronoun, as in I like her; and case ii) deals with the instances where her is a possessive - I like her new hat.

The trees become



(12)

Nothing has been assigned to the right hand branch of the analysis tree, because prediction 4, the only result of this line of analysis, failed.

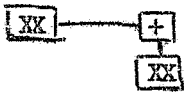
Is can only be a singular verb (again it is the function of the closed-class dictionary to tell us this), so that, referring to the current prediction tree in diagram 12, all predictions except number 1 will fail,

with the result that they and their successors are removed from the tree.

Unlike an ordinary verb, however, the verb to be does not have the possibility of taking no, one, or two objects. Instead, simplifying as extravagantly as ever, we may pretend that to be must be followed by one and only one complement. On this assumption, it is easy to see that the assignment of)s to is results simply in the replacement of)s on the prediction tree by = , the complement sign. When, therefore, the word is in our example has been processed, the prediction tree will have collapsed down to the single set of predictions

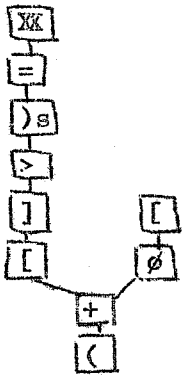
<- (13)

The last word of our sentence, handsome, is an unmarked open-class item, and may, by the rules of the algorithm, be marked with the complement sign = . The effect of assigning this mark to such an item is exactly the same as if the mark were > ; the mark is removed and its successors are promoted to make one set of immediate predictions, and the mark is replaced by the continuation symbol + to make another immediate prediction. Thus immediately before the full stop in our present example is read, the prediction tree is



<- (14)

The full stop is now encountered, and only the left hand prediction succeeds. Since we last displayed the analysis tree in diagram 12 we have assigned to it in succession the marks)s , = , XX . At the end of our pass through the sentence, therefore, the complete analysis tree looks like this:



.
handsome
is
her
kisses
who
sailor
the

(15)

and the sentence has a unique analysis, namely the one shown at the beginning of the example:

(the + sailor [who] kisses > her)s is = handsome XX .

It will be true in general that any sentence which is marked with more than one pair of brackets will contain more than one deep structure. (The reverse is probably, but not necessarily, true also.) It does not follow that because a sentence has an unambiguous surface structure it will also have unambiguous deep structure. For instance, our surface structure analyser may discover that a particular sentence contains a relative clause; but it gives no indication of which noun the clause is appended to. Thus, in an example such as I saw many things at the house which I had never noticed before, the surface structure is probably unambiguous, but we cannot be really certain whether the relative clause belongs to things or to house.

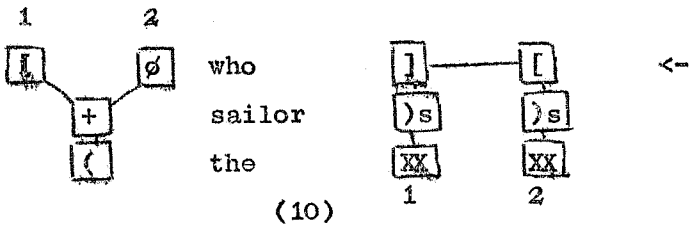
In the example we have just worked through, the interplay between the words of the sentence being analysed and the prediction tree is becoming more evident. We have seen how the marks assigned to a word indicate its position in the syntactic structure of the sentence, and now we have also seen a couple of examples of the way in which a particular word may modify the prediction tree in a particular way, reflecting the roles which we know that the word is capable of playing. In our full algorithm, words such as the and who make more important and complicated changes to the prediction system than we have shown them making here, but nevertheless, we have given some idea of the way in which their grammatical possibilities are all taken into account.

Finally, let us consider as an example a sentence very similar to the one we have just analysed. Written out with the analysis we shall derive for it, it is

(the + sailor ø who [she]r married)s is = happy XX .

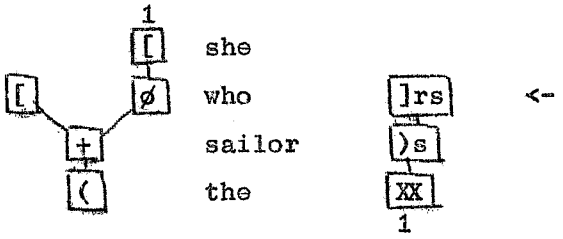
This sentence contains the second kind of relative clause mentioned

above, where the relative pronoun is an object of a following verb. (Pedantry would insist on 'The sailor whom she married ... ', but we are sure that, at any rate in spoken English, the form with who is by far the more common.) We shall skip briefly through the analysis of the example, starting immediately after the word who has been analysed, when the analysis and prediction trees are in the state shown above and repeated here as diagram 10.



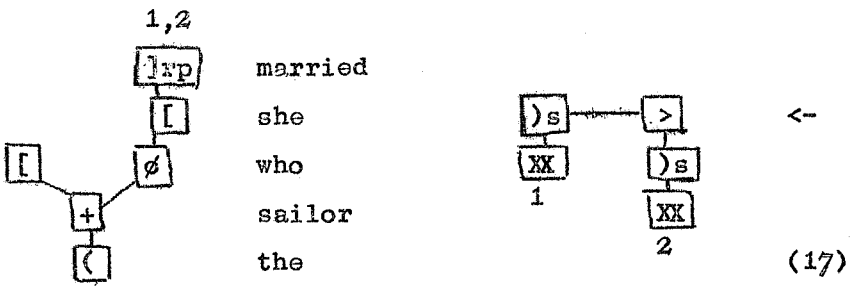
(10)

The left hand prediction fails, as the closed-class dictionary indicates that she cannot be a verb. The right hand prediction succeeds, however, and we note that if [is applied to she, then the corresponding] must be marked singular. The new prediction and analysis trees are now formed. (Remember that a suffix r is necessary on the].)



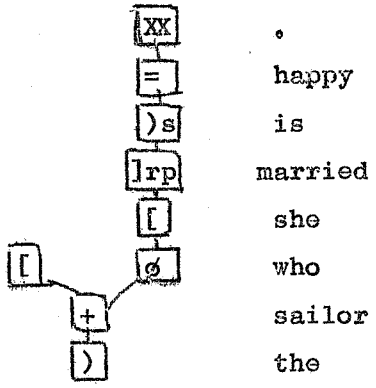
(16)

Married is an open-class item marked by a terminal -ed, so that it may perfectly well be a singular verb, and the only immediate prediction, of]rs, is therefore successful. It is now that the suffix r is taken into account, for on assigning]rs to married only the predictions no objects or one object are made for this verb, and not the usual no, one, or two objects. The two trees therefore become



(17)

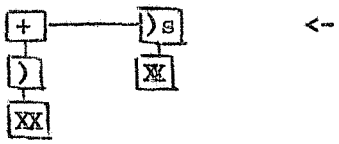
The analysis of the rest of the sentence proceeds almost exactly as in the previous example, until we end up with the analysis tree



(18)

showing that there is one analysis of the sentence, namely, that written out at the beginning of the example.

Now the important point about this sentence is that we do it very little damage by omitting the word who. The sailor who she married is happy and The sailor she married is happy manifestly have almost identical surface structures. If, however, we tried to analyse the second of these two sentences using the simplified algorithm we have described so far, we would arrive at the word she with the following prediction tree (taken from diagram 2)



(19)

and it is at once apparent that she cannot be marked with either of the immediate predictions, as it can neither continue a noun phrase (*a blonde she) nor be a verb. Thus the algorithm would fail to analyse the sentence, which would be a pity.

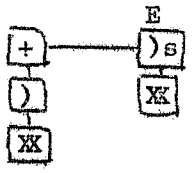
We therefore say that, in certain circumstances, the mark [, that is the start of an embedded subject, may be self-predicting. The term, while perhaps something of a misnomer, is nonetheless sufficiently clear, the implication being that a situation has been reached which was always possible, but which we had no grounds for predicting at any particular moment. Another way of solving this problem would be, of course, simply to add each of these 'possible but we don't really know whether it will happen now' predictions on to the list of immediate predictions in the usual way, but in fact we prefer the system with self-predicting marks for two reasons. The first is simply that the amount of computation necessary is less if we do things like this. The second, and more important reason, is that it seems on the whole to be a better model of human behaviour. For example, one noteworthy class of self-predicting items is 'time-words'. If we hear a sentence which starts <u>If you see Fred tomorrow</u> ..., it seems to us unreasonable to say, or to imply in our model, that we in any sense predict the word <u>tomorrow</u>. It was certainly always possible that it would come along, and we would not be checked in our analysis of the sentence by encountering it, but it was really not a possibility that we bothered about until we actually heard the word. This is very different from the way in which, having heard <u>If you</u> ..., for instance, we are definitely predicting the occurrence of a verb for this clause.

Returning to the case of the self-predicting embedded subject, we see that, unlike a time word, say, this cannot occur at almost every position in a sentence, but only in certain positions. Very roughly, when a noun phrase is ended, the possibility of embedding is 'turned on', and when next a mark is assigned, the possibility is 'turned off' again. (In fact the question is very much more complicated than this, and the possibility of embedding depends in a complex way on, among other things, the determiner at the beginning of the noun phrase.) We may express this possibility in our much-simplified examples by adding E immediately above those immediate predictions which may be preceded by a self-predicting [. (Notice that not all predictions are forced to be in the same E/not E

state, since we may, for instance, have derived some of the immediate predictions by assigning a verb mark to a particular word, thus removing the possibility of embedding, while the other immediate predictions may be the result of treating the same word as a noun, leaving the possibility of embedding open.)

The possibility of embedding or other self-predicting forms is controlled (in our full algorithm) by a collection of flags associated with each immediate prediction, and these flags are among the factors which have to be taken into account before two immediate predictions of the same mark can be conflated.

If we now return to our example The sailor she married is happy, we see that the prediction tree after the word sailor has been read, allowing now for embedding, will look like this:

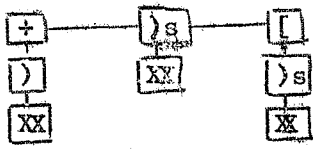


<-

(20)

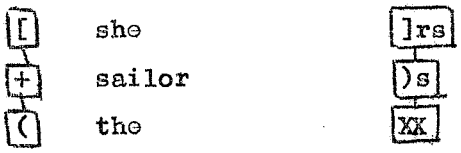
The left hand immediate prediction arose from considering the sailor ... to be the first two words of an incomplete noun phrase. On this assumption, therefore, the noun phrase has not yet ended, so that no embedding is possible. On the other hand, the right hand immediate prediction arose from taking sailor to be the last word in its noun phrase. A noun phrase has therefore been finished, no mark has subsequently been assigned, and so embedding is possible.

Now when she is read, an attempt must be made to assign to it not only the marks + and)s, but also [, just in case it turns out to be the start of an embedded subject. In this case, of course, only the [fits. Assigning a self-predicting [has exactly the same effect on the two trees as if the mark had been predicted in the normal way by a branch of the prediction tree which had [for its immediate prediction, and all the marks with a superposed E for its successors. That is, the case above works just as if (20) had in fact been



<- (20a)

Thus, after [has been assigned to she, the analysis and prediction trees are



<- (21)

From here on the analysis of the rest of the sentence follows exactly as before, so that we end up with a unique analysis

(the + sailor [she]sr married)s is = happy XX .

(An observant reader may have noticed that this is not, strictly speaking, the first time we have encountered self-predicting marks. When we discussed who earlier, we said, among other things, that it was to be marked with either [or ø, even though, in fact, neither of these marks was available on the prediction tree as an immediate prediction. It will now be recognised that we were able to mark who thus simply because it is, in effect, the first word of a 'self-predicted' embedded clause.)

This must conclude our brief account of the algorithm we use for analysing the surface structure of an input sentence. It cannot be stressed too often just how over-simplified such an account must necessarily be. Many important features of the algorithm have not been mentioned and no attempt has been made to describe exactly even those few features which have been mentioned. Most important of all, no effort has been made to describe the second-stage deep-structure analyser.

III

There is an interesting extension that can be made to the program. For the reasons given above words like boy and laugh are not entered in the dictionary employed in the analysis procedure, which means that no form class information is supplied by the dictionary look-up routine for these words other than that they are open-class items. But using information derived from the analysis the program can, so to speak, learn that boy is a noun and laughed a verb and can, in effect, construct for itself another dictionary - an open-class dictionary - in which this information is stored. This can be shown in an oversimplified fashion as follows. Take the sentence The boy laughed. The dictionary look-up routine translates this into the string Definite Article, Open-Class Word, Open-Class Word - the being the only word contained in the closed-class dictionary. A search of the syntax with which the analyser is supplied indicates that the sentence must comprise at least a noun phrase and a verb phrase and that one of the expansions of Noun Phrase is Definite Article + Noun. This, in fact, is the only one which could have figured in the generation of this particular sentence since any rewriting of Noun Phrase involving more than two symbols would leave no symbol for the rewriting of Verb Phrase. From this it follows both that boy is a noun and laughed a verb - a hypothesis that is confirmed by the discovery that the latter incorporates a verbal inflection.

Notice that in many instances during the early stages of running the program it is inevitable that incorrect entries will be made in the open-class dictionary. Given the sentence the cat adores fish, two analyses will be produced - the desired analysis and one in which the cat adores is taken as a noun phrase on the analogy of phrases like the boy scouts. Adores will therefore be entered tentatively both as a verb and a noun. But after a while an automatic correction routine can be run on the open-class dictionary which, for example, discovers any word which has been entered as being both a noun and a verb but for which, while there have been unambiguous instances of its being labelled as a verb, no cases

have been found in which it has been labelled as a noun without at the same time an analysis being produced in which it has been labelled a verb. In this case the dictionary entry for the word is modified by the deletion of the label noun. If at a later stage the same sentence is submitted for analysis two analyses will again be produced. But now both analyses can be checked against the open-class dictionary the program has itself constructed. If taking the sentence as ambiguous means treating one of the words as a part of speech different from that the dictionary records it as belonging to, this is sufficient reason for dropping this analysis. Given an analysis of a sentence in which every word functions as the part of speech as which it usually functions, we are unlikely also to recognise another analysis for the sentence in which one of the words now functions in an entirely unexpected way. For example, no one is likely to take the sentence John laughed as an imperative, on the analogy of a sentence like Bring water, since this would involve taking John as a verb while there is a perfectly acceptable analysis of the sentence in which John functions in the normal way as a noun. By making the analytic procedure and the open-class dictionary arising from it interact in this way, as the open-class dictionary using the information produced by the analysis improves so too will the analyses produced.

Following out this procedure it is possible not only that the program can learn that boy is a noun and laugh a verb, but that laugh is an intransitive verb. But if the program is to acquire all the information the English speaker has about these words, it is necessary also that it should know that boy, for example, is a concrete noun and an animate noun. It is possible that this kind of syntactic information too might be automatically derived. For this to happen, however, it would first be necessary for information about syntactic features actually to be supplied for certain words. Say, for example, we include in the original closed-class dictionary the word surprise plus the information that it is a verb that must always take an animate noun as its object. Then when the sentence The boy surprised the teacher has been analysed the program will

learn not only that teacher is a noun but also that it is an animate noun. If the next sentence to be analysed is The teacher laughed, it will now learn that laugh is the kind of verb that can take an animate subject. In this way the original information concerning syntactic features can be spread over the whole lexicon. Clearly there are many problems here. For example, many verbs can take animate, inaminate and abstract subjects, and the fact that up to a certain point the program has not encountered an instance of a verb taking one type of subject is no guarantee that it cannot do so. Moreover it is by no means clear which verbs, or how many verbs, or even whether it is verbs at all, that one should choose as the starting point. Nevertheless this looks like an interesting field for experiment.

Behind the idea of automatic dictionary construction lies the idea that except perhaps in the clear unambiguous cases, like the grammatical formatives, the process of assigning syntactical information to words cannot usefully be compared to looking them up in a dictionary: that no matter how many times we may have heard the word boy on every occasion we work out what part of speech it is by analysis. (This, of course, does not preclude the possibility of our being surprised when we compare the results of one analysis with the previous results of analysing the same word.) We must assume that the speaker has a complete knowledge of the language he speaks - that is, that he has internalised a complete grammar of that language. But in discussing the way in which he deploys this knowledge in analysing sentences it is not necessary to assume that it includes rules which rewrite terminal symbols by words. Indeed it might even be misleading to suggest that it does.

Footnotes

(1) It might be argued that since the input to the device is written rather than spoken sentences it would be better described as a model of the reader rather than of the hearer. The fact remains, however, that there is no reason to suppose that the way in which we employ our knowledge of the syntax of our language in listening to a sentence differs from that in which we employ that knowledge in reading a sentence. On the other hand there are no grounds for equating the orthographic form of the sentence with its form after phonological processing, or for assuming that the phonological analysis of a sentence must be completed before syntactic analysis can begin. There is, in fact, evidence that suggests that all the phonological information associated with a sentence is available only after a syntactic analysis has been imposed upon it.

(2) For a full account of the Harvard Predictive Analyser see Murray E. Sherry,'Comprehensive Report on Predictive Syntactic Analysis', Mathematical Linguistics and Automatic Translation (Report No. NSF-7, Harvard Computation Laboratory, Section 1, 1961).

(3) See G. H. Mathews, 'Analysis by Synthesis', 1961 International Conference on Machine Translation of Languages and Applied Language Analysis (H.M.S.O., 1962), 531-539, and 'Analysis by Synthesis in the Light of Recent Developments in the Theory of Grammar', Proceedings of Colloquium on Algebraic Linguistics and Machine Translation held at Prague, September 1964 (forthcoming).