AN AUGMENTED STATE TRANSITION NETWORK ANALYSIS PROCEDURE

Daniel G. Bobrow
Bolt, Beranek and Newman, Inc.
Cambridge, Massachusetts

Bruce Eraser
Language Research Foundation
Cambridge, Massachusetts

Summary

A syntactic analysis procedure is described which obtains directly the deep structure information associated with an input sentence. The implementation utilizes a state transition network characterizing those linguistic facts representable in a context free form, and a number of techniques to code and derive additional linguistic information and to permit the compression of the network size, thereby allowing more efficient operation of the system. By recognizing identical constituent predictions stemming from two different analysis paths, the system determines the structure of this constituent only once. When two alternative paths through the state transition network converge to a single state at some point In the analysis, subsequent analyses are carried out only once despite the earlier ambiguity. Use of flags to carry feature concordance and previous context information allows merging of a number of almost identical paths through the network.

Introduction

The"present paper is an abbreviated account of a syntactic analysis system, STAP, (State Transition Analysis Procedure) implemented In BBN-LISP (Bobrow, Murphy, Teitelman 1969) which is designed to take as its input a sentence of English written in normal orthography and produce as its output a form of deep structure representation of the sentence. We require that the analysis of an Input sentence contain the information present in the transformational deep structure representation of a sentence, making explicit such relationships as logical subject, logical object, etc.

This is not a novel enterprise; reports of work with similar goals can be found in papers by Petrick (1965), MITRE (Zwicky, et. aL, 1965), Kuno (1965), and Thorne, Bratley, and Dewar (1968). The first three actually produce a tree structure which purports to be a linguistically motivated deep structure in the form like that described by Chomsky (1965). Thorne, Bratley and Dewar attempt to capture this same information by appropriate labelling of nodes, and insertion of some dummy nodes in a surface structure analysis. The Fctrick, MITRE, and kuno analysis algorithms depend explicitly on a transformational grammar of some cannonical form. The analysis algorithm refers to this grammar during the analysis of an input string. The Petrick and MITRE approaches are distinguished from that of kuno In that the former explicitly involve transformation reversal while the latter, once having found the surface structure representation of the sentence, goes directly to the deep structure.

It was after reading the paper by Thorne, Dewar and Bratley that we became convinced that an efficient and linguistically interesting analysis system could be developed without explicit reference to a transformational grammar. It is, in fact, the Thorne work which provided us with the general structure of STAP: the notion of a state transition network for reflecting immediate grammatical restrictions and the use of flags for carrying information relevant to other parts of the analysis. STAP is an elaboration of the procedure described by Thorne, Bratley, and Dewar (op^ cit.); the authors wish to gratefully acknowledge the generous time and assistance provided to us by Thorne and Dewar which permitted us to understand their system and thus attempt STAP.

The goals of these two systems differ: Thorne et. al. are concerned with the development of a psychologically valid model of syntactic analysis utilizing a limited dictionary (basically only function words) and a single pass algorithm. In STAP we are primarily concerned with reflecting those deep structure relationships motivated within the framework of a transformational grammar and utilizing a complete dictionary with fully specified entries. In addition STAP can transmit information back Into a previously analyzed structure, a capability not provided in the Thorne system. The STAP system has the full

formal power of a Turing machine; an important and nontrivial question is whether the linguistic information in either the Thorne or STAP system can be introduced by some simple function operating on a transformational grammar such as one from the class of grammars described by Petrick (op. cit.). This question has not yet been carefully investigated but we anticipate an affirmative answer.

In this paper we attempt to characterize clearly and precisely (1) the form and content of our analysis of an input sentence; and (2) the nature of the algorithm that is utilized. In no sense does this analysis algorithm provide any semantic information, either of the sort found in Katz (1966) or that used by Woods (1968) as an interface to his system for querying a structured data base (in his case the Official Airlines Guide). That is, no linguistically or procedurally motivated semantic Information is contained in the analysis. However, since both the Katz and Woods approaches to the semantic interpretation of a sentence utilize a deep structure representation equivalent to what STAP produces, the STAP analysis could serve as a first for such semantic analyses.

The General Structure of STAP

Superficially, STAP resembles a finite state automaton. The analysis algorithm, starting at an initial state S0, uses each word of the input string to cause a change of state through a state transition network. If, at the end of the input string, the system is in the final state, END, the Input has been analyzed as a well-formed sentence. The history of the transitions provides an analysis of the structure of the sentence. In short, the general strategy of the analysis procedure can be viewed as a series of state transitions with each transition signifying some additional successful subanalysis.

However, a strict finite state automaton with transitions dependent only on the next word is well-known to be inadequate as a model for the analysis of a natural language. STAP obtains additional power from the capability of the transition interpreter to save its current status and call Itself to analyze a substring by starting anew at some other state In the network. This recursive ability of STAP provides it with the power of a non-deterministic push-down store automata with a top-down parsing strategy; thus it is capable of recognizing an arbitrary context-free language with indefinite embedding of sentences.

However, STAP goes beyond the ability of the numerous context-free parsing algorithms already implemented to account for a range of linguistic information contained in the deep structure analysis of a sentence. This capability of STAP derives from the auxiliary operations performed during each state transition, and additional information which STAP associates with the constituents of the analysis. The operations include the setting of flags, the testing of previously set flags to enable or reject a transition, and the transferring of information determined by the current transition to some previous constituent(s) of the analysis.

The State Transition Network

There are two types of transitions between a pair of states of the state transition network. The first, the more straightforward, is where a single word satisfies the input conditions and permits the state transition to occur. The second is where the next substring of the input must be analyzable as some constituent structure (e.g. a noun phrase) for the state transition to be permitted. We refer to these two types as lexical and structural transitions, respectively. In (1) we have shown an illustrative state transition network for STAP.

The subpart in (II), the top level, indicates that an input string must be analyzed as a structure, S, followed by a terminal punctuation mark, ("." or "?") That is, we have first a structural transition and then a lexical transition.

In order to permit the structural transition from S0 to /S it is necessary for STAP to save its current status and to reenter the network at the state S, shown in (Iii). The transitions from the state S to the state C-LIJD (constituent end) as shown in (Iii) specify the possible analyses for the constituent S. For example, a sentence can be analyzed as NP-AUX-VP (e.g. John can see Mary) requiring the transitions from S to /NP to /AEX to C-END; or analyzed as SADV-NP-VP, requiring transitions from S to /ADV to /NP to C-LND (e.g. Certainly the man is dead).

However, the transition from S to /NP is of the structural rather than the lexical type, and requires that STP save its status and reenter the system

anew, this time at the state NP shown in (liii). Reentering the network in the way we have been describing is equivalent to going down a level in the constituent structure; that is, separating the constituent being analyzed into its dominated constituents (e.g. UP into DET-ADJ-N).

We note here that there are certain syntactic categories such as NP and AUX which are satisfied by both a lexical item or a structural constituent. For example, the input word it_ is analyzed as a NP, and so is the string DET-ADJ-N (the big man). Such cases are combined in the network into a single transition which is possibly both structural and lexical.

The Analysis Procedure

Two alternative strategies for parsing exist within the system. The choice between them depends in part on whether one wants to obtain one analysis or all possible analyses for each input sentence. For the single analysis case, at each node in the state transition network STAP can try (in some suitably arranged order) each of the possible transitions until one succeeds, proceed to the state designated, and continue in this way until the input string has been completely examined. If in any state Sj no transition is possible, STAP must back up to the previous state Si_ , mark the transition from Si_ to Sj as blocked, and then try some other transition from Si. Left recursive grammar rules pose no difficulty if the transition which Implies the left recursion is tried last. To find all analyses, STAP can remember a successful analysis at the state END, and then simulate a failure. It will then recursively traverse the network through all possible paths until no untried transitions are available from any reached state. With this technique, great care must be taken with left recursion to prevent infinite loops.

We are interested in obtaining all analyses, but do not use the recursion technique just described. Rather, STAP simulates the operation of a non-deterministic pushdown store, and simultaneously follows all possible transitions from a given state. Thus at any time in the analysis of a sentence, all partial analyses up to a particular input word are available (though not necessarily with all of the final information associated with the constituents) and a number of paths through the network are active. All and only those paths which stay active through the last word of the input sentence and finish in the state END represent successful analyses.

In order to deal with the case of common predictions at a point in the input string (of which left recursion is a special case), whenever STAP must reenter the system during a structural transition, a check is made to see if this subconstituent has been expected previously at just this point in the Input string analysis. For example, after analyzing through the word flying In the ambiguous sentence They are f]ylrig planes, there would presumably be "just two analyses, both predicting that the next constituent could be a noun phrase. Accordingly, the analysis of the MP is carried out only once.

A number of auxiliary techninues are used to *compiless* the size of the state transition network by storing information in other ways. For example, since only one sentence adverbial (SADV) Is allowed per sentence, the two transitions shown In (1) requiring, SADV cannot both be allowed nor can more than one transition around the loop at state /NP be permitted. We implement restrictions of this type by providing with each successful transition a capability for setting flags. A parallel facility allows tests for appropriate flags before a transition is permitted. In the present example, either of the state transitions requiring a sentence adverbial causes a flag to be set indicating that an SADV had been analyzed at this level. In addition, a condition for the possiblity of such a state transition Is that this SADV flag has not been previously set. Thus, the first time a sentence adverbial is analyzed, the setting of the SADV flag precludes any further sentence auverblal analysis. The use of a flag as discussed thus allows compression of the network in the form shown as opposed to requiring two separate paths, one in which the sentence adverbial is in the initial position, the other in which it is in the pre-AUX position. We note here that the setting of the SADV flag also carries information which will preclude the analysis of the sentence as a question.

Another set of transitions which are differentially allowed by flags are those shown for VP. Both VTR (transitive verb) and VIN (intransitive) go to the same state, /V, but in the former case a noun phrase must follow while in the latter case it cannot. Thus we associate with the VIN transition the setting of a flag which precludes the state transition /V to /IO.

We also allow a further flag setting mechanism based on the subcateg^-orizatlon of a word class. For example, the verb sleep has the subcategorizatlon features [+V, +Voluntary Action, - _NP] associated with it in the dictionary. In analyzing this verb, various flags are set based on this information. For example, the syntactic feature -_NP sets a flag, call it NODO (no direct object), which is tested by the /V to /IO transition. The NODO flag set precludes this transition. To consider another example, to distinguish between a single object verb such as hit and a double object verb such as give set another flag. Both the state transitions in (Ilv) from /IO to /IO in analyzing to and from /IO to C-END analyzing a noun phrase test for this flag and are permitted only if it is set.

In addition to the preceding mechanisms, STAP transfers information backwards to previously analyzed parts of the sentence. For example,, when the transition from the state /IO to itself occurs in the analysis of the preposition to it is possible to state definitely that the noun phrase analyzed In the /V to /IO transition is functioning as the direct object. As a part of this transition, the information Indirect Object is added as a comment to the constituent NP immediately following the verb. Furthermore, if > this transition has not occurred, then the /IO to C-END state transition through a noun phrase derives the information that this noun phraee is functioning as the direct object, and that the noun phrase immediately following the verb is functioning as the Indirect object. This functional Information is assigned to the relevant constituents as a result of this transition. The actual information transfer situation is of course more complex because one or both of these objects may have been moved to another position in the sentence. We emphasize again that it is this type of information transfer technique that permits the claim that STAP can capture the deep structure information associated with a sentence.

An alternative technique for determining the analysis of a constituent is to make alternate predictions when the functional status or other information relevant to a constituent is not clear. In the double object case, for example, STAP can make two alternative predictions concerning a noun phrase following a verb like give/, one that the noun phrase was the direct object; the other that it was the indirect object. STAP then supresses the analysis which turns out to be incorrect. The choice between these two mechanisms is dependent in each instance on the relative costs of maintaining extra analyses as opposed to the cost of searching back through the present analysis to find the appropriate place to store the Information once it is derived. For a sentence like Who should the book have been given to yesterday? there are a number of active alternative analyses just after processing the noun phrase the book: one must allow who to be the direct or indirect object; and the book to be the direct object, the indirect object for a passive construction, or the subject of an active verb. Some of these alternatives can be eliminated when the determination is made (after the have been given) that a passive sentence is being analyzed. Only after the preposition tc has been analyzed is it possible to determine that the who is functioning as the Indirect object
anci
the book as the direct object.

In order to compress the state transition network, we allow state transitions which do not require using any of the input words. For example, the transition labelled NIL in (Iv) from state /N to C-END allows termination of the constituent AUX after just one aspect (ASP) word or one modal (M), or after each cycle through the ASP loop. Another such transition is the one labelled (NP) from /IO to C-END in (liv). This transition is used to insert into an analysis a marker indicating where the indirect object position was in the deep structure in a sentence such as Which person did you show it to? A transition of the same type is used to mark the deep structure position of that part of the auxiliary which has been moved forward to form a question. (See (6) below).

The Form of the Analysis

The goal of STAP is to determine for a given input sentence a deep structure analysis which reflects the basic logical relationships of a sentence necessary for its semantic interpretation. (The surface structure representation reflects only the order and superficial relationships among the grammatical constituents of the sentence as produced by the speaker.) Ambiguous sentences, of course, have more than one deep structure. Quite clearly, a natural language analysis system cannot be put to any interesting use in the area of artificial intelligence unless the Information present in the deep structure representation is

determined.

Basically there are three types of information in the deep structure analysis of a sentence: (1) categorial information specifying what syntactic categories are present and their hierarchical relationship; (2) functional information such as what constituent is functioning as the logical subject of the sentence, which as the direct object, etc., and (3) subcategorization information reflecting finer specification of the constituents present (the complex symbols as characterized by Chomsky, 1965, e.g. that the noun cat is subcategorized as animate, non-human). however, in order to specify this information especially (1) and (2) above, the deep structure analysis usually does not resemble the surface structure in a variety of ways. The two most crucial differences between the deep and surface structure analysis of a sentence are (1) the order of t'he constituents in the one relative to the other, and (2) the additional structure present in the deep structure not found in the surface structure analysis. For example, a simple sentence such as The red dog is noisy has a deep structure ana~lyYis"paraphrased roughly by The__dog which is red Is noisy where the relative order of "reel ana dog are reversed and a relative clause is present.

In STAP, if the relative position of a constituent in the deep structure analysis is the same as in the input string (the surface structure), this is explicitly stated in our analysis. For those cases where the relative order is altered, this constituent is marked in the surface structure position as out of place and its original position in the deep structure is indicated. In short, the general form of our analysis resembles the surface structure analysis of the sentence, with added indications of moved constituents and where they are located in the deep structure. Co-referentiallity is indicated by appropriate labelling.

We will illustrate our analysis using the relatively complicated sentence (2) Was the man believed to have been shot by John and present its surface structure analysis (3), its usual deep structure analysis (4), and its STAP deep structure analysis (5). The form of the output is three columns: the first containing the categorial information; the second the functional information; and the third the subcategorial information. The first and third are usually combined into a tree (P-marker) in the linguistic literature but we have found the present adaptation more workable and equally as readable. It is important to recognize that the functional information associated with the surface structure analysis (3) is that determined from the surface structure itself and is usually referred to as the grammatical as opposed to the logical functional relationships (Cf. Chomsky, 1965, for a discussion of these terms.)

Let us now trace through the analysis of this sentence to see what the system does to produce the modified deep structure analysis (5) associated with (2), ignoring the numerous blind alleys which the analysis routine must follow. Lssential to understanding this procedure Is the notion that each constituent has only a few canonical forms in the deep structure, (e.g. S may be ADV-NP-AUX-VP or NP-AUX-VP; VP may be V, V-NP, V-NP-PP; etc.) and these facts are built into the state transitions in the same way as they are characterized by the phrase structure rewriting rules of the base component of a transformational grammar. Thus, in terms of the analysis already performed and the present state, the system is always trying to meet one or more of these cannonical formats.

Looking now at (2), the analysis of was as an AUX is not an acceptable initial constituent for a deep structure S and causes the marker # to be placed after AUX signifying that it is not in Its deep structure position relative to the other constituents in the sentence. The string the man is then analyzed as a NP, but its function is left unspecified (e.g. subject or object). The next word, believed, is a verb (according to the dictionary of STAP) either of past tense or past participle form. The earlier presence of the was precludes the past tense analysis and furthermore indicates the passive construction. The past participle analysis of believed requires it to be the first constituent of a VP (we are ignoring the reduced relative clause analysis here, e.g. the man (who was) believed to be ill.); at this point a flag is set to enable a transition within this current VP (believed to have been shot by John) to accept a logical subject of this sentence in the form by-NP. The subject NP-AUX sequence preceding this VP, are introduced into the analysis by

allowable null transitions. Each is followed by a * signifying that the actual constituents have been moved elsewhere.

The dictionary entry for believe provides subcategorization information which sets a flag which requires that this constituent be followed by an NP. An NP in STAP may consist of an N, a DET-N sequence, a DET-N-S sequence, or simply an S. The presence of the to following believe eliminates all but the S analysis of the required following NP. In reentering the system at the S level, STAP provides content information depending on what has already been analyzed indicating that the type of S we can anticipate is a declarative, not an imperative or a question. Moreover, to enter the S state after the verb requires an NP, or a dummy NP which has been moved elsewhere; thus, the NP-* is introduced.

At this point the system "knows" that the topmost sentence has been passivized with the grammatical subject (the man) coming from an embedded sentence in the deep structure. At issue now is the deep structure analysis of this embedded sentence. The fact that the first ASP, have, is followed by a second, been signals that this embedded S is also in the passive form. Thus, when the verb shoot is encountered, beginning a VP, the system again expects to find a by-NP which functions as the logical subject of the sentence someplace within this VP. In addition, following the verb shoot is an NP-* which represents the deep structure position of the logical object of this embedded sentence. In this example, by John immediately follows the verb, satisfies the agent requirement, and the analysis of the embedded VP is completed.

At this point in the system the embedded S transition has been satisfied and the analysis of the top S continues; the passive agent has yet to be accounted for. Since there are no more words in the input string, the system "assumes" that this subject NP has been deleted and provides the appropriate analysis, namely, an indefinite NP (represented by someone in the linguistic literature.) If the example in (2) were ambiguous between John doing the believing or the shooting, then a second analysis of the sentence would be identical to that Just described except that the embedded S would have the deleted NP and the top S would have the by John associated with it.

The reason we believe that a reasonable deep structure grammar can be written in this form is that there is very little permissible distortion to an underlying cannonical form that a particular S can endure. From a linguistic point of view, the number of transformations which can apply to a particular S is relatively small, usually only two or three. Moreover, for each type of distortion—movement of constituents to the right or left, permutation of constituents, deletion of constituents, and various combinations of these—it appears to be possible to determine not only what has occurred but to reconstruct the deep structure analysis.

For example, extraposition of the that-S sequence to form the sentence It is obvious that Henry is mad from the deep structure order It that Henry is mad is obvious is signaled by the sentence initial It. The initial it doesn't require that extraposition has occurred but must be present if it has. Similarly, with sentences such as There seems to be something rotten around here. Left movement of an NP such as in the passive construction in relating The man was seen by everyone with Everyone saw the man is verified as soon as the was-past participle signifying the passive construction is analyzed. That the grammatical subject (e.g. the man) need not be the logical subject of the top level sentence is clear[1] from the example sentence in (2) and the subsequent discussion and we have illustrated how its correct deep structure analysis can be effectively handled. The logical subject in the passive case, if present, Is of course signaled by the preceding by. Permutation such as in questions usually results In an order of constituents not included in the list of deep structure cannonical structure analyses. These cases appear to be the easiest to recognize and deal with since the constituents involved are adjacent both before and after the permutation. Deletion poses a different sort of problem since what is deleted is usually not simply a word but an entire phrase.

A case such as verb phrase complementation where the deep structure subject NP and AUX have been deleted is analyzed in Figure 6. Note that the required deep structure co-referentiality between the direct object of persuade and the subject of go (the NP, John) is Indicated by a generated name NP-1 used as identical subcategorial features associated with the two constituents. The embedded NP reflects none of its structure in our analysis; it is

actually redundant since the first object NP-1 contains the identical information.

Conclusion

We have described a syntactic analysis system which obtains the deep structure information associated with an input sentence. We believe that for the analysis of a sentence to be useful in Artificial Intelligence this level of analysis must be available. The implementation utilizes a state transition network characterizing linguistic facts representable in a context free form, and a number of techniques to coae ana derive additional linguistic information and to permit the compression of the network size thereby allowing more efficient operation of the system. by recognizing iaentlcal constituent predictions stemming- from two uifferent analysis paths the system can determine the structure of this constituent only once. This both increases the efficiency of the system operation and permits use of left recursive context free rules. When two alternative paths through the state transition network converge to a single state at some point in the analysis, subsequent analyses are carried out only once despite the earlier ambiguity. Use of flags to carry feature concordance and previous context information allows merging of a number of lamost identical paths through the network.

References

1. Bobrow, D., b. Murphy, and w. Teitelman. (196b) "BBN LISP System.'' Bolt, Beranek and Newnian Inc., Cambridge, Mass.

2. Chomsky, N. (1965) ''Aspects of the Theory of Syntax.'' M.I.T. Press, Cambridge, Mass.

3. Katz, J. (1965) "The Philosophy of Language.'' Harper and Row, New York.

H. Katz, J. and P. Postal. (19660 "An Integrated Theory of Linguistic Description." M.I.T. Press, Cambridge, Mass.

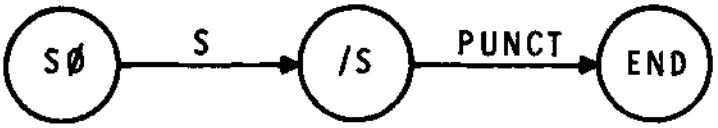
5. Petrick, S. (1965) "A recognition Procedure for Transformational Grammars." Unpublished Doctoral Thesis, M.I.T., Cambridge, Mass.

6. Thome, J., P. Bratley, and li. Dewar. (1968) "The Syntactic Analysis of English by Machine." In Michie, D (Ed.) "Machine Intelligence 3." American Elsevier Press, New York.

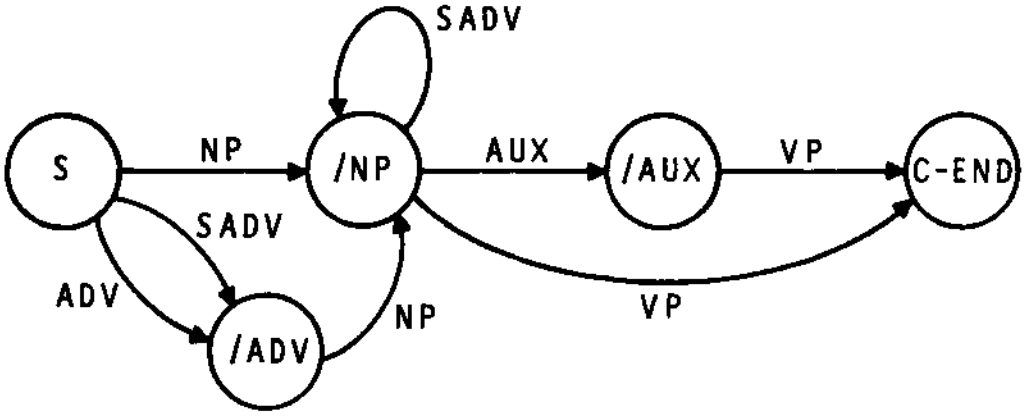
7. Woods, W. (1968) "Procedural Semantics for a Question-Answer Machine.'' In the Proceedings of the Fall Joint Computer Conference, Thompson Book Company.

(1) AN ILLUSTRATIVE STATE TRANSITION NETWORK

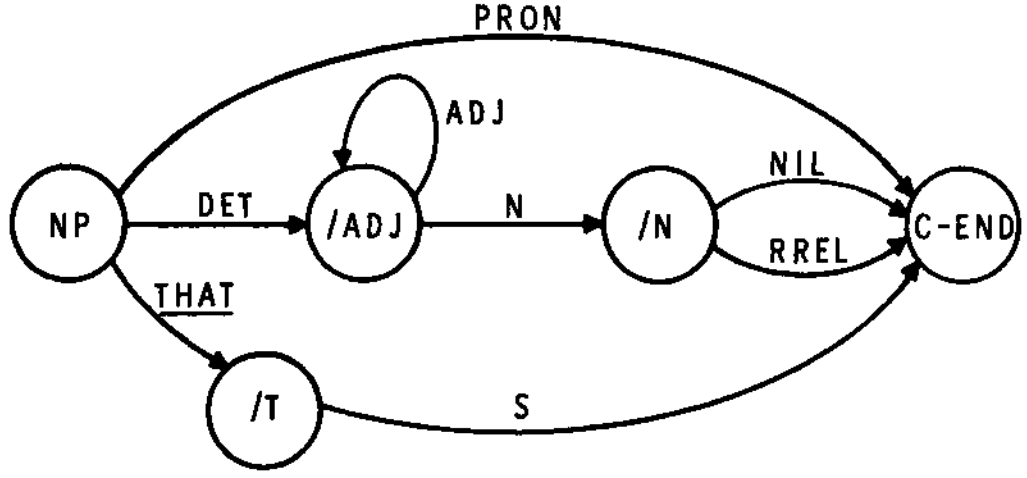
(i)



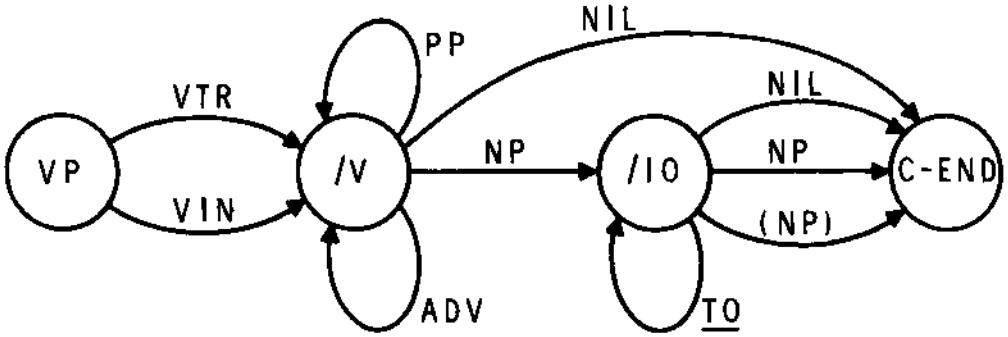
(ii)



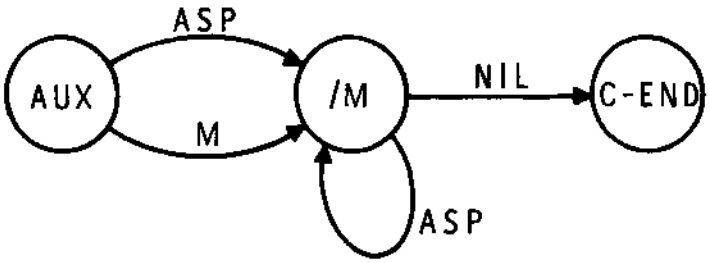
(iii)



(iv)


VP
VTR
VIN
/V
PP
ADV
NP
/IO
NIL
NP
(NP)
TO
NIL
C-END

(v)


AUX
ASP
M
/M
NIL
C-END
ASP

(3) Surface Structure Analysis

Categorial	Functional	Subcategorial
S	QUESTION	
AUX		
ASP-was	SUBJECT	[PAST]
NP		
DET-the		[SPECIFIC]
N-man		[COUNT; SING]
VP		
V-believed		[STATIVE; COGNIT]
to		
AUX		
ASP-have		[PRES]
ASP-been		[PASTPRT]
VP		
V-shot		[PASTPRT; ACTIVE]
PP		
P-by		
NP		
N-John	PREPOBJECT	[PROPER]

(4) Deep Structure Analysis

Categorial	Functional	Subcategorial
S	QUESTION	
S		
NP	SUBJECT	
N-Ø		[INDEF]
AUX		
TNS-Ø		[PAST]
VP		
V-believe		[STATIVE; COGNIT]
NP	COMPLEMENT	
S		
NP	SUBJECT	
DET-the		[SPECIFIC]
N-man		[COUNT; SING]
AUX		
TNS-Ø		
ASP-have		[PRES]
VP		
V-shoot		[ACTIVE]
NP	DIRECT OBJECT	
N-John		[PROPER]

(5) STAP Deep Structure Analysis

Categorial	Functional	Subcategorial
S	QUESTION	
AUX-#		
TNS-Ø		[PAST]
ASP-be		
NP-#	OBJECT	[SPECIFIC]
DET-the		[SPECIFIC]
N-man		[COUNT; SING]
NP-*	SUBJECT	
AUX-*		
VP		
V-believe		[STATIVE; COGNIT]
NP	OBJECT	
S		
NP-*	SUBJECT	
AUX		
ASP-have		[PRES]
ASP-been		[PASSMKR]
VP		
V-shoot		[ACTIVE]
NP-*	OBJECT	
PP		
P-by		
NP-#	SUBJECT	
N-John		[PROPER]
PP		
P-by-Ø		
NP	SUBJECT	
N-someone-Ø		[INDEF]

The symbol # after a constituent indicates that it is not its deep structure position; the symbol * indicates that the constituent so marked was here in the deep structure but has been moved elsewhere; the Ø indicates deletion.

(6) We persuaded John to go to school.

Categorial	Functional	Subcategorial
S	STATEMENT	
NP	SUBJECT	
N-we		[PRON; PLR]
AUX		
TNS-Ø		
VP		
V-persuade		[ACTIVE]
NP	DIROBJECT	[NP-1]
N-John		
S	COMPLEMENT	
NP-Ø	SUBJECT	[NP-1]
AUX		
VP		
V-go		[ACTIVE]
PP		
PREP-to		[DIRECTION]
NP	PREPOBJECT	
N-school		[INANIMATE; SING]