AN ATN TREATMENT WH-MOVEMENT

Hans Haugeneder
Siemens AG
ZT ZTI
Otto-Hahn-Ring 6
8 München 83, West Germany

ABSTRACT

An ATN-Parser is presented with emphasis on the treatment of those phenomena which in the framework of transformational grammar are subsumed under the concept of WH-movement. The approach taken tries to embed these constructions into an ATN grammar in a general, linguistically motivated and in terms of the ATN grammar formalism descriptive way. To accomplish this goal the approach described incorporates the basic principles governing such constructions as formulated in the framework of the trace theory proposed in the development of the Extended Standard Theory (EST). Thus a unified treatment for both relative clauses and wh-questions is achieved.

1 Overview

1.1 The Grammar

The ATN-Language used is - except for some minor deviations which are not of importance here - in accordance with the one specified in [BATES 78] and covers the following subset of English:

- the most frequent verb types for declarative and imperative sentences and questions
- direct and indirect WH-questions (for NPs, PPs, ADJPs, ADVPs)
- direct and indirect Y/N-questions
- sentential complements for verbs and nouns
- various types of relative clauses (complete, reduced...)
- infintive construction (including control verbs with subject and nonsubject control)
- conjunction of complete constituents of every syntactic category
- subordinate clauses

The grammar is written in very compact way, making extensively use of merging techniques; it comprises about 75 states and 170 arcs.

The structure built by the grammar is quite similar to the phrase structure trees used in transformational grammar on the S-structure level. (See appendix for some examples!)

Structures of that type seem to us very suited for further semantic interpretation, since they offer a level of syntactic structure which both expresses grammatical functions (which are defined configurationally) and constituent structure features which are important for certain aspects of logical form as e.g. quantifier scoping. Especially the use of the device of traces (as a phonologically empty phrase) allows for displaced elements to appear in the phrase marker in their surface postion and at the same time expresses the role which this element's original position plays in the corresponding predicate-argument structure.

1.2 The Parser

The parser itself is an active chart parser as described in [KAPLAN 73] and [KAY 73]; a detailed desciption of our implementation can be found in [ENDERS et al 82].

By combining an active chart, which represents all fully analyzed (sub)constituents (the passive edges) and all incomplete partial derivations (the active edges) with an agenda, which contains an explicit representation of all further tasks to be processed, the chart parsing framework is especially suited for mul-ti-way analyses on syntacticly and lexically highly ambiguous input.

Furthermore it offers a high degree of flexibility in the use of various control-structures beyond the uniform ones like depth-first and breadth-first. This can be achieved by means of heuristic measures of various types which are used for the weighting of the single tasks, thus being responsible for the ordering of the agenda. Thereby several features of the parser such as closure and attachment features and preference of certain readings not induced by attachment ambiguities can be varied effectively by simply modifying this weigthing function.

There exist three versions of the parser differing in the strategy by which the grammar is processed:

- a top-down version (which is the standard case with ATNs)
- a top-down version augmented with an automatic one-word look-ahead on PUSH-arcs

- a mixed bottom-up/top-down version

The latter two versions which both demand some pre-rocessing of the grammar are clearly better in terms of performance than the pure top-down version. They have about 75% of the memory requirements and need 80% of the cpu-time of the top-down version.

All three versions have been implemented in Siemens-Interlisp running on Siemens main-frames. An implementation on a lisp machine is on the way.

2 Integrating WH-Movement into ATN Grammars

2.1 The Descriptive Adequacy of ATNs

ATNs, though being a (computationally) powerful specification language for grammars are not committed to a certain lingistic theory as stated cleary in [WAHLSTER 79]. This often leads to grammars in which various types of syntactic phenomena are treated descriptively inadequate and linguistically unfounded. In the case of the wh-movement phenomena being discussed here, one has to cope with the fact that the origin of the displaced element can be embedded in some deeper clausal constituent arbitrarily far away from its surface position. (For an example see figure 2. in the appendix !). Sticking to the standard facilities offered by ATNs for constructions of that type implies the use of SENDR actions on PUSH-arcs or the HOLD/VIR mechanism in order to allow the involved subnets to communicate with each other. [1]

But neither of the two possibilities can be viewed as a satifactory solution. The former (pushing certain register contents through possibly several subnets) burdens the grammar writer with the tasks of controlling the use of the pushed register either by using it in some network or by pushing it further down. It thus puts extreme emphasis on the procedural aspect of the grammar thus making it clumsy and hard to comprehend. Therefore this approach can hardly be considered as a satisfactory solution to the problem.

The second possibility, namely the use of a global register environment and virtual arcs has been introduced into the ATN framework in order to handle displaced elements of the sort discussed here. In this mechanism the HOLD-action offers the facility to put an analyzed constituent on the HOLD-list (a global additional stack). Later on a VIR-arc can consume a constituent from the HOLD-list as if it occured at the actual point in the input string.

[1] The necessity of the use of one of these two mechanisms is thereby only under the assumption, that the phrase structure to be produced is thought to express the functional role played by an displaced element in some way or the other.

Thus the HOLD/VIR facility and extensions of it as proposed in [FININ 83], though surely being preferable to the first possibility still misses to express the relevant syntactic restrictions and permits violations of grammatical constraints governing these constructions.

Thus to us it seems worthwhile to incorporate wh-phenomena into ATN grammars on a conceptually higher and syntactically motivated level.

2.2 Description of the Approach

2.2.1 WH-Movement in EST

Before decribing the embedding of WH-movement into an ATN grammar a short presentation of the general principles of its treatment in EST is given.

The fundamental feature of it is the use of an transformation (move wh-phrase) which moves a wh-phrase of a nonverbal phrasal category into an sentence-initial non-argument position by adjoining it to the COMP-node. The moved phrase leaves behind an empty phrase (its trace), which is coindexed with the fronted wh-phrase.

This fronting can be achieved in one step as in the unbounded movement analysis or successivly as in the COMP-to-COMP analysis, yielding the same resulting structure in both cases (ignoring the intermediate COMP-dominated traces). The version reflected in our approach is the unbounded movement analysis as only the COMP-node finally dominating the wh-element is affected.

2.2.2 The ATN-Treatment

2.2.2.1 General Idea

The basic idea in the proposed treatment of wh-construction is to let the parser build up phrase structures containing unbound traces and bind these traces to the fronted wh-phrase when the entire wh-construction is accomplished.

This task itself is divided into the following two steps:

1. The phrase structure trees built during the parse may contain unbound traces. At the grammar level this is achieved by adding JUMP-arcs to the grammar, which can accept empty constituents. This means that you have JUMP-arcs parsing traces of type XP, which in the grammar are alternative arcs to the corresponding PUSH XP-arcs (with XP other than VP). Furthermore these arcs contain certain register actions responsible for building the structures for the empty phrases.

   The possible locations of traces as subconstituents in the wh-construction phrase structure depend on the restrictions described by these JUMP-arcs when the grammar is applied to the input.

2. On the level where the phrase structure trees for wh-constructions are constructed a binding procedure tries to bind (i.e. coindex) the fronted wh-phrase with its trace.

Since this binding procedure is actually the central part of the mechanism its features are described in more detail in the following section.

2.2.2.2 The Binding Procedure

What the binding procedure (BindWhTrace) actually does is to establish a mapping of a phrase structure tree into another phrase structure tree, where the mapping is structure preserving in the sense that it does not alter the phrases' internal structure. The effect of the mapping is to establish a coindexing between a wh-phrase in COMP-position and its trace in an argument position as shown in the following diagram.

(S/(COMP(ADJP < +wh >)..)(S..(ADJP e)..))

==>

(S/(COMP($ADJP_i$ < +wh >)..)(S..($ADJP_i$ e)..))

The coindexing itself is restricted by various constraining conditions. These concern the morpho-syntactic, functional and configurational features of the two phrases to be bound.

So, for example the wh-NP "whom" in COMP-position can only be coindexed with an empty NP bearing the grammatical function of direct object or prepositional object for example. Furthermore there are configurational constraints between the two candidate phrases that have to be fullfilled, in order to establish a proper binding as e.g. the coordinate structure constraint. This constraint does not allow movement of a phrase out of a conjunctive structure which, being applied to the binding procedure means that the coindexing may not take place in a structure where ß is an empty constituent of the category XP'.

(S/(COMP (XP' < +wh >).)..(XP(XP...ß..) and (XP..).)

not ==>

(S/(COMP(XP'_i < +wh >).)..(XP(XP..$ß_i$..) and (XP..).)

Since the phrase structure built at the point where the binding procedure is involved may contain traces which are no suitable candidates for a proper binding (since they do not fullfill these restricting conditions) the procedure may not succeed to establish a proper indexing. That's why the procedure can also be viewed as a function reporting success or failure respectively, thus filtering out part of the structures built so far.

Furthermore even in the case of a proper binding done by the binding procedure the resulting phrase structure trees may still contain unbound traces. Such structures however must be blocked according to two constraints which do not allow constituents in wh-question and relative clause structures to be moved outside the entire phrase, which in terms of the coindexing means that they may not be coindexed with a constituent outside these structures.

In the case of (headed) relative clauses this is ruled out by the complex-NP constraint, which (stated in a simplified form) does not allow a constituent ß to be moved out of the following structure:

(NP...N/(S/...ß...))

For questions an equivalent restriction is expressed by the wh-island constraint, which does not allow a phrase ß to be moved out of the following structure:

(S/(COMP.. < +wh >)(S...ß...))

Since the binding procedure does its work just on this sort of phrase structure, namely the structures produced by the relative clause and the wh-question subnets, it can easily test these two constraints too, thus blocking all structures with no proper binding of the displaced wh-element or with leftover unbound traces.

Now the way in which this procedure is embedded in the grammar should be clear. It has been incorporated into each POP-arc of a wh-construction subnet (i.e. the relative clause and the wh-question subnet), where it is used as a test-predicate reporting whether the structure built so far contains a proper binding with no unbound traces left. The actual binding is done as a side-effect of that predicate. So the corresponding POP-arc look like this:

(POP pm' (BindWhTrace pm)) [1]

(With pm' derived from the phrase marker pm by

---

1 In terms of the grammar used, a POP-arc for a wh-construction network has the following form:

(0.7 POP (GETR R)(BindWhTrace R))

(Here R is the register containing the phrase structure tree on which the binding procedure works. The first item in this arc is a weight assigned to it.)

successful application of BindWhTrace)

Thus POP-arcs with an additional test of the sort described act as filters popping the form pm' only if this test has been sucessfully applied to the structure pm.

2.2.3 Interaction with the Treatment of Control Verbs

As already mentioned our grammar also deals with certain contol verb constructions. These are also handled by a coindexing procedure which functions similar to the binding procedure. It is invoked on the POP-arc of the corresponding network and tries - based on the lexical features of the control verb in the matrix clause - to coindex the abstract subject "Pro" with its controller, also reporting success or failure.

In cases where this coindexing mechanism and the binding procedure may affect the same constituent (as an example see figure 2. of the appendix) we have adopted the strategy of doing the wh-element binding before the control-indexing. This means that the controller-NP may already have assigned an index that actually is used for the control indexing of Pro. So in our example the final structure is constructed in the following way:

S'(COMP(NP < +wh >).)(S.VP(V persuade)(NP e)(S'.(S(NP Pro).))))

wh-binding ==>

S'(COMP(NPi < +wh >).)(S.(VP(V persuade)(NPi e)(S'.(S(NP Pro).))))

control binding ==>

S'(COMP(NPi < +wh >).)(S.(VP(V persuade)(NPi e)(S'.(S(NPi Pro).))))

This reflects the ordering of these two mechanisms in transformational grammar (namely the move-wh transformation and the rules of construal) ; thus the correct handling for a wide variety of structures involving both wh-movement and control constructions has been achieved.

2.3 Comparison to other Approaches

2.3.1 Wehrli's GB-Parser

In this section the comparison with other parsers is limited to the discussion of the handling of wh-constructions only; thus this is not intended an overall comparision of the parsers per se.

One approach to the treatment of wh-constructions that lends itself well to a comparison can be found in Wehrlis's parser for French [WEHRLI 83]. There a binding procedure is incorporated as a separate module in a parser which in toto is based on the GB-framework [CHOMSKY 82].

The procedure is invoked at the same level, namely when a wh-construction is actually parsed (i.e. whenever the syntactic rule is applied, which combines a wh-element with a phrase of the category S.) This roughly corresponds to the POP-arc of a wh-construction subnet in our case.

But in contrast to our approach where the binding procedure acts on a constituent structure already containing traces, Wehrli's binding procedure is activated by the absence of an obligatory slot in the functional structure. After locating such a slot an empty element is inserted both into the functional and the constituent structure with a suitable index. So in his approach the search for the argument position traces (which in our approach is done by the parser by offering different phrase structures already containing traces) is part of the binding procedure.

This, on the one hand has the advantage of focussing on the treatment solely in the binding procedure. On the other hand, it makes the procedure considerably more complex, since it must be able to detect ambiguities [1] which in our approach are dealt with in the grammar rules.

2.3.2. PARSIFAL

The second approach discussed here is the work by Marcus on wh-movement in his PARSIFAL system [MARCUS 80]. There the cyclic analysis of wh-movement is adopted which, besides its linguistic motivation [CHOMSKY 77] is enforced by the structure of the grammar interpreter. The reason for this lies in the fact that during the parsing of some embedded clause S_e there is no access to any structure beyond this S_e as at that time it is the current cyclic node in the active node stack. So for example in a structure like

(S(COMP (NP' < +wh >))...(S_e ...(NP" e) ...)...)

at the moment when the trace NP" is parsed (i.e. created, dropped into the buffer and attached to S_e)

there is no access to the headed wh-NP NP' thus preventing the trace NP" to be bound to it. This

1 There may be several unfilled slots in the functional structure, in the matrix sentence as well as in some embedded sentence. This is for example the case if the headed wh-phrase is a PP whose attachment is ambiguous, as in:

On which day will John hold a speech?

(with its "when"-reading "When will ...." and its "topic"-reading "On which topic will .... ")

problem is avoided, if the grammar rules create a COMP-Node with a bound trace attached to it before analyzing S_e. Thus the trace NP" can be bound to the headed wh-NP via some intermediate COMP-attached traces, as for example via NP''' in the following structure:

(S(COMP (NP'$_i$ < +wh >))..(S$_e$(COMP(NP'''$_i$ e)..(NP'' e).).)

This mechanism is implemented by means of a special register, WH-comp, which in must be handled explicitly the grammar rules when creating and binding traces. Furthermore the status of that register (which can be set in the grammar to "utilized" and "not utilized") serves as a condition for triggering certain actions in the course of the anlysis of wh-constructions (such as the deactivation of certain rule packets). This from the grammar writer's point of view seems to have some of the flavour and disadvantages of the SENDR-solution discussed before (section 2.1).

The approach described so far however causes difficulties in a productive class of constructions, where the trace to be analyzed is dominated by an NP which also serves as the current cyclic node making the headed wh-phrase inaccessible for the binding of its trace as in constructions like:

(S/(COMP(NP who))(S is that (NP a portrait of e)))

This problem is overcome by Marcus by means of an ad hoc mechanism which exactly allows access to a headed wh-element in such constructions, whereas in our approach constuctions of both types are handled in a uniform way.

3 Conclusion

The approach described has from our experience proved to be a useful step in the direction of handling wh-movement constructions in an ATN grammar in a linguistically motivated way. The resulting grammar turned out to be more perspicuous than a grammar dealing with these phenomena with the standard ATN facilities (as sketched in in section 2.1).

The work described will be continued in three directions. First, we will incorporate this treatment of wh-constructions and other syntactically motivated mechanisms (as for example NP-movement) in a grammar for a substantial subset of German. Secondly, we will include further binding-mechanisms for various kinds of anaphoric relationships. Thirdly, we will explore the possibility of embedding such concepts into the ATN language formalism.

ACKNOWLEDGEMENTS

The implementation of the various versions of the parser and the grammar editor has been done in cooperation with my (partly former) colleagues Reinhard Enders, Ludwig Schmid and Bertram Fronhöfer.

REFERENCES

[BATES 78]
Bates, Madelaine, "Theory and Practice of Aug-mented Transition Network Grammars", in Bolc, Leonard (ed), *Natural Language Comunication with Computers*, Berlin, 1978

[BENNIS & GROOS 82]
Bennis, Hans, and Groos, Anneke, "Die Rektions-Bindungs-Theorie: Neue Aspekte seit den 'Aspekten' ", *Zeitschrift für Sprachwissenschaft*, 2 (1982)

[CHOMSKY 77]
Chomsky, Noam, "On Wh-Movement", in Culicover, P. W., Wasow, Thomas, and Akmajian, Adrian (eds), *Formal Syntax*, New York, 1977

[CHOMSKY 82]
Chomsky, Noam, *Lectures on Government and Binding*, Dordrecht 1982

[ENDERS et al 82]
Enders, Reinhard, Fronhöfer, Bertram, Haugeneder, Hans, and Schmid, Ludwig, *Überlegungen zu einem ATN-Chart-Parser*, Internal Report (Siemens), 1982

[FININ 83]
Finin, Timothy W., "An ATN Programming Environment", in Bolc, Leonard (ed), *The Design of Interpreters, Compilers, and Editors for Augmented Transition Networks*, Berlin, 1983

[GÖRZ 79]
Görz, Günther, "ATN und Kontrollstrukturen", in Christaller, Thomas, and Metzing, Dieter (eds), *Augmented Transition Network Grammatiken*, Berlin, 1979

[KAPLAN 73]
Kaplan, Ronald, "A General Syntactic Processor", in Rustin, Randall (ed), *Natural Language Processing*, New York, 1973

[KAY 73]
Kay, Martin, "The Mind System", in Rustin, Randall (ed), *Natural Language Processing*, New York, 1973

[MARCUS 80]
Marcus, Michell P., *A Theory of Syntactic Recognition for Natural Language*, Cambridge/MA, 1980

[WAHLSTER 79]
Wahlster, Wolfgang, "ATNs und die semantisch-pragmatische Steuerung der Analyse und Generierung natürlicher Sprache", in Christaller, Thomas, and Metzing, Dieter (eds), *Augmented Transition Network Grammatiken*, Berlin, 1979

[WEHRLI 83]
Wehrli, Eric, "A Modular Parser For French", in *Proc. 8th IJCAI Karlsruhe 1983*, Los Altos/CA, 1983

APPENDIX A. Sample Parses

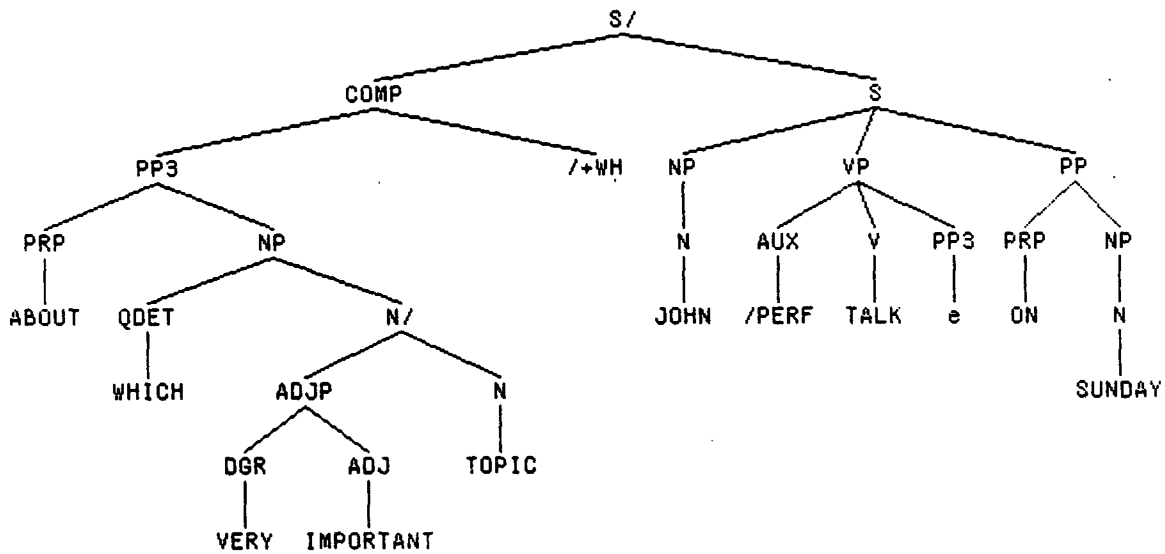

Figure 1. Parse tree for " About which very important topic has John talked on Sunday?"

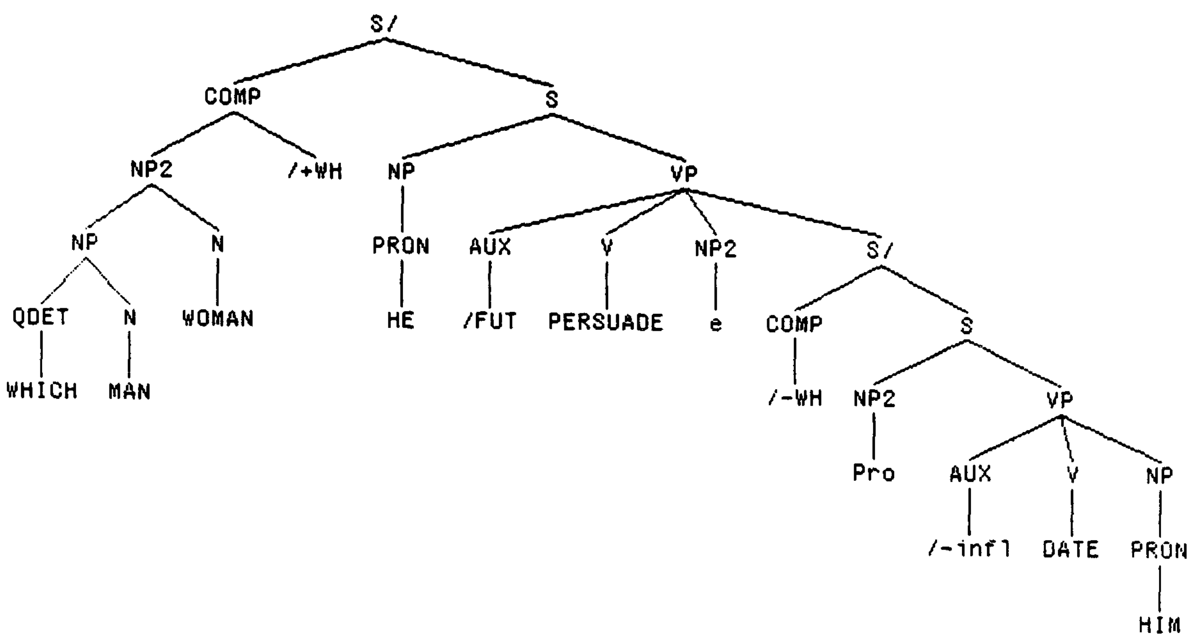

Figure 2. Parse tree for " Which man's woman will he persuade to date him?"

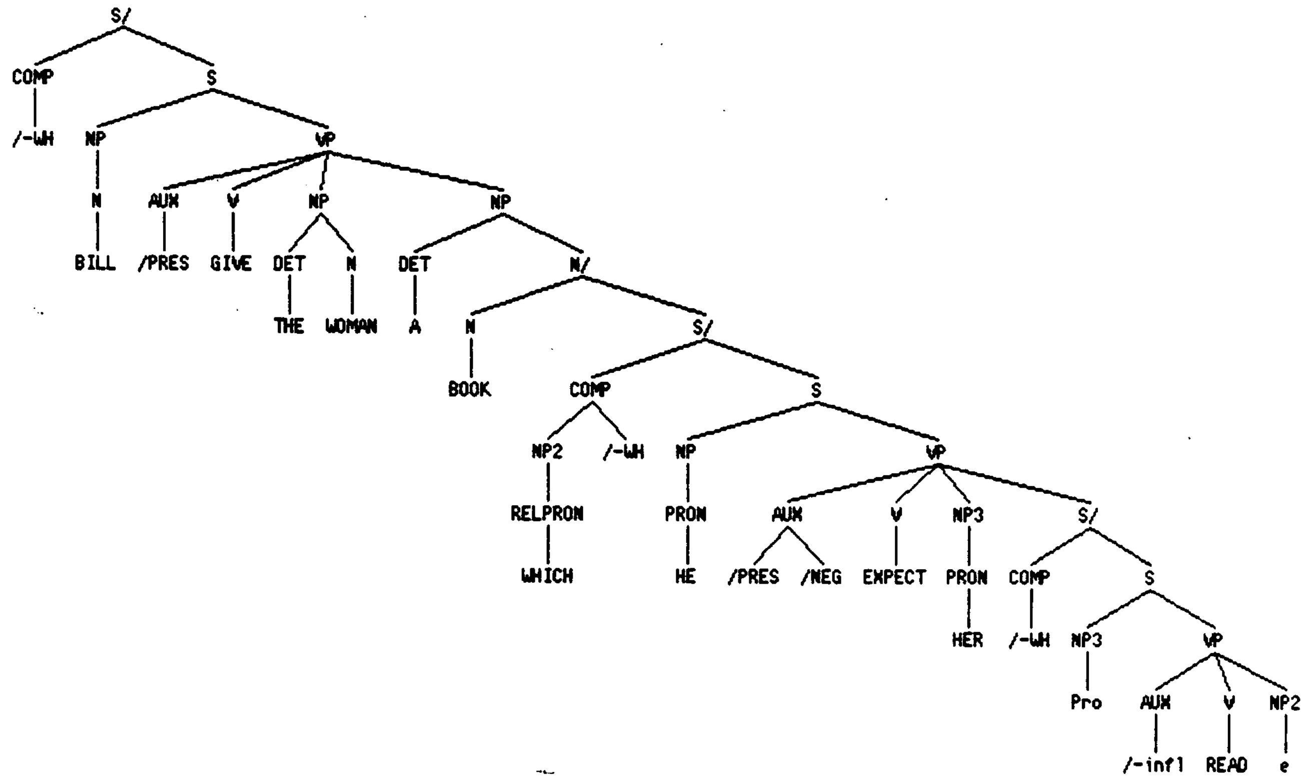

Figure 3. Parse tree for " Bill gives the woman a book which he does not expect her to read."