Seven principles of surface structure parsing in natural language*

JOHN KIMBALL

Indiana University

Abstract

In generative grammar there is a traditional distinction between sentence acceptability, having to do with performance, and sentence grammaticality, having to do with competence. The attempt of this paper is to provide a characterization of the notion 'acceptable sentence' in English, with some suggestions as to how this characterization might be made universal. The procedure is to outline a set of procedures which are conjectured to be operative in the assignment of a surface structure tree to an input sentence. To some extent, these principles of parsing are modeled on certain parsing techniques formulated by computer scientists for computer languages. These principles account for the high acceptability of right branching structures, outline the role of grammatical function words in sentence perception, describe what seems to be a fixed limit on short-term memory in linguistic processing, and hypothesize the structure of the internal syntactic processing devices. The operation of various classes of transformations with regard to preparing deep structures for input to parsing procedures such as those outlined in the paper is discussed.

1. Introduction

In grammar there is a distinction between those sentences which are rejected by speakers on grounds of grammaticality versus those rejected for performance reasons. Thus, there is a quadripartite division of the surface structures of any language: (a) Those sentences which are both grammatical and acceptable, e.g. 'It is raining';

* I am indebted to Frank DeRemer for many discussions on the material in this paper, particularly for providing information on current computer work in parsing, and for suggestions concerning the principles discussed in section 4; also to Cathie Ringen, Jorge Hankammer, Paul Postal and the reviewers for reading and commenting on an earlier version.

(b) those sentences which are grammatical but unacceptable, e.g. 'Tom figured that that Susan wanted to take the cat out bothered Betsy out'; (c) those which are ungrammatical but acceptable, e.g. 'They am running'; and (d) those which are both ungrammatical and unacceptable, e.g. 'Tom and slept the dog'.

Part of the problem facing linguists, in particular psycholinguists, is to find a characterization of the processing of linguistic experience, i.e. performance, adequate to distinguish unacceptability from ungrammaticality. In the following paper I will present an interconnected set of principles of surface structure parsing in natural language which is intended to provide a characterization of the notion 'acceptable sentence'. In particular, these hypotheses will explain the difficulty experienced by native speakers of English when sentences of the classical problem types such as in (1) are encountered.

(1) a. That that that two plus two equals four surprised Jack astonished Ingrid bothered Frank.
    b. Joe believes that Susan left to be interesting.
    c. The boat floated on the water sank.
    d. The girl the man the boy saw kissed left.

Further, these hypotheses will explain why a sentence like (2b) is not normally interpreted as (2a) in the same way that (3b) can be interpreted as (3a).

(2) a. The woman that was attractive took the job.
    b. The woman took the job that was attractive.

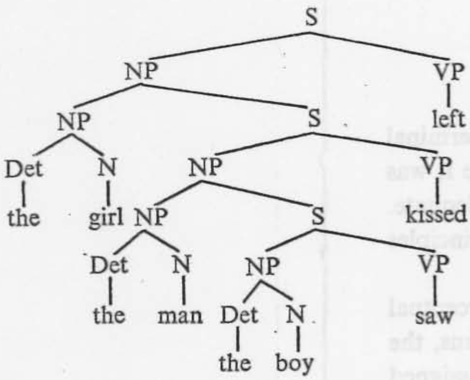
(3) a. The woman that was attractive fell down.
    b. The woman fell down that was attractive.

The first section of the paper will be concerned with the traditional hypotheses that have been presented to account for the unacceptability of sentences like (1). In Section 3 I will look at some of the parsing techniques designed by computer scientists for processing the sentences of programming languages. Of particular interest will be those techniques which allow a string to be parsed top-down left-to-right building a phrase structure tree over the string as it is read, since this is the process employed by speakers of natural languages. We will find that there are restrictions on the class of languages which allow this kind of parsing, and it will be possible to examine the claim that part of the function of transformations is to put deep structures in a form to allow limited memory left-to-right parsing. It will also be possible to examine the claim (*cf.* Chomsky, 1965, chapter 1) that transformations function to construct a surface tree which is optimally parsed by those techniques available to native speakers. In Section 5 we will see that the known transformations of English divide themselves into three distinct classes with respect to the kinds of output structures that are produced.

2. Some previous accounts of surface structure complexity

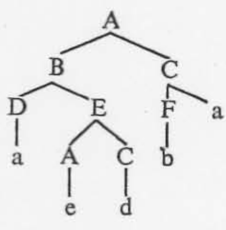
It is interesting to consider first attempts to explain the low acceptability of a sentence like (1d) 'The girl the man the boy saw kissed left'. The surface parse tree for this sentence is shown in (4).

(4)

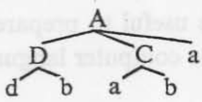


One hypothesis (Fodor and Garrett, 1967) is that such sentences are perceptually complex due to the low proportion of terminal symbols (words) to sentences. In general, one of the effects of transformations seems to be to reduce the non-terminal to terminal ratio in surface structures (Chomsky, 1965) as compared to deep structures; i.e., to map relatively tall structures such as (5a) into relatively flat structures such as (5b).

(5) a.



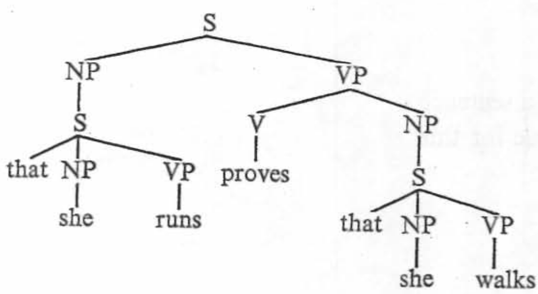
b.



It is argued that flat structures are preceptually less complex, although no explanation of this fact has been offered. Why it should be that flat structures are easier to parse than deeply embedded structures is accounted for by Principle Two of Section 4 below.

The Fodor and Garrett hypothesis seems to be a specific form of this more general hypothesis. This specific hypothesis is, however, easily falsified by considering sentences which have fewer terminals per S than (1d), but which are perceptually much less complex. (6) is such a sentence.

(6)



In (1d) the ratio of terminals to S-nodes is 9:3; in (6) it is 7:3. Thus, simple terminal to S ratio will not account for all cases of perceptual complexity, even those it was designed to cover. The more general terminal to non-terminal ratio is also inadequate. That this ratio is relevant to perceptual complexity can be predicted by the principles presented below in Section 4.

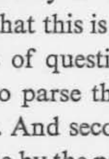
Chomsky and Miller (1963) claim that (1d) is complex because the perceptual strategy of assigning a N (noun) to the following V (verb) is interrupted. Thus, the subject-verb relation must be assigned to *the boy* and *saw* before it can be assigned to *the man* and *kissed*, and this before it can be assigned to *the girl* and *left*. This hypothesis treats the difficulty of center embedding as one of assigning grammatical and, perhaps, semantic relations to the elements of a sentence. The principles discussed below treat the difficulty in (1d) in terms of surface tree configuration only. I conjecture, then, that it is not the impediment of a principle of semantic interpretation which is involved in the complexity of (1d). Rather, the complexity lies in the structure of the surface tree. The principles below attempt to outline the exact nature of this difficulty. Before presenting these principles, it will be useful to prepare the way for them by considering techniques of parsing designed for computer languages.

3. Parsing algorthms for programming languages

Programming languages are often based on context free languages; the problem of designing parsers for such languages is solved by constructing a parser from the CF grammar for the language. Since the parser 'knows' the productions of the lànguage, its job is then defined as that of selecting which productions had to have applied to generate a particular input string. (A usual requirement for programming languages is that their grammar be unambiugous, so the solution to the parsing problem for any string is always unique.) The parser must reconstruct the derivational history of a string from context. For example, if a grammar contains two productions with identical right hand sides, say $A \rightarrow X$, and $B \rightarrow X$, where X is an arbitrary string of terminals and non-terminals, then when the parser reads X, it must decide from context which production in fact applied. For example, *A* might be introduced by a rule C → D A, D → d, and *B* by the rule C → E B, E → e. Thus, when X is to be parsed as an *A*, it will always be preceded by a *d* and when it is to be parsed as a *B*, it will be preceded by an *e*.

Given that this is the general type of problem that must be solved by a parser, then two types of questions may be asked concerning a particular language. First, is it possible to parse the string deterministically from left-to-right as it is read into the computer. And second, what is the size of the largest context that must be examined at any time by the parser to decide how to build the parse tree? For reasons of efficiency, an optimal programming language can be considered to be one in which it is possible to parse left-to-right, and for which there is a fixed finite bound on the size of the forward context which must be examined at any point.

There are two general strategies used in parsing algorithms (*cf.* McKeeman *et al.*, 1970). In the first, a tree is built for an input string by starting with the initial symbol of the grammar (that which is topmost in all trees generated by the grammar) and building a tree downwards to the terminal symbols. Such procedures are called *top-down*. The operation of such a procedure on an input string $a_{i_1} \ldots a_{i_n}$ for a grammar with an initial symbol *S* is illustrated below:



The first step of the algorithm is to build a tree down to the first symbol of the input string, a_{i_1}. The language must be such that it is always possible to do so uniquely by examining no more than *k* symbols ahead of a_{i_1} (where *k* is fixed for the language) if the parser is to be deterministic. At this point, *B* acts as the new 'initial' symbol for some substring, and the parser operated to complete *B*, again working top-down. When B is filled out, the next higher node in the pathway down from *S* is the new initial symbol, for the purposes of the parser. The class of grammars which permit parsing by an algorithm such as that outlined above is called the class of LL(k) grammars. For such languages trees can be built top-down, and it is never necessary to look more than *k* symbols ahead of the given input symbol to determine what action is to be taken by the parser.

The second type of parsing procedure involves building a tree from the bottom-up. The first action of such a parser is to assign the first *m* input symbols to some node, which is then placed at the top of a stack. Thus, in parsing $a_{i_1} \ldots a_{i_n}$, if the first *m* symbols are dominated by *B*, the parser will operate as illustrated below:

	Stack	*Input string*
Stage 1:		$a_{i_1} \ldots a_i a_{i_{m+1}} \ldots a_{i_n}$
Stage 2:	B	$a_{i_{m+1}} \ldots a_{i_n}$

The parse is completed when the initial symbol is the only symbol in the stack, and the input string has been completely read in. Those languages which can be parsed bottom-up by looking ahead in the input string no more than *k* symbols are called LR(k) languages; the grammars which generate such languages are called LR(k) grammers (*cf.* Knuth, 1965).

In general, parsing of computer languages differs from that in natural languages in two significant ways. The first involves the fact that programming language grammars are unambiguous; a parser for a programming language yields a unique tree for each string. Not only this, but also the behavior of the computer parser must be deterministic. That is, its action at any given string of input terminals and stack configuration must be uniquely determined. A model of parsing in natural language must allow for more than one parse and should predict on the basis of considerations of surface structure complexity which parse will most likely be offered as the first choice by a native speaker. Second, and most important, parsing in computer languages differs from that in natural languages in that a computer parser is allowed an essentially unrestricted memory. For example, in the case of a parser for an LR(k) grammar, it is possible to look ahead *k* symbols and then decide that the appropriate action is to read in the next symbol. This can be done until the whole string is read into memory before the parse tree starts being built. On the other hand, there is considerable evidence that short-term memory (STM) in humans is quite restricted, and that a tree must be built over an input string constantly so that the initial parsed string may be cleared from STM. Principle Four below concerns what seems to be a fixed limit on linguistic STM. Because of the limitations on STM, the form of a parseable (acceptable) surface structure in natural language is quite restricted.

4. Six or seven principles of surface structure parsing

The following principles, although presented here as distinct, are closely linked and interact in various ways, as will be pointed out during the discussion. The central principle of the scheme discussed here is the last, and from it various of the others follow deductively.

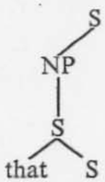
4.1 *Principle One (Top-Down): Parsing in natural language proceeds according to a top-down algorithm*

The operation of such algorithms was outlined above in Section 3. The claim made is that parsing of natural languages is like that in LL(k) languages, with one variation noted below. Thus, the first node built in such parsing is the top S, while in bottom-up parsing this is the last node to be constructed. Before considering the consequences of such an assumption, let us consider it in operation in parsing a sentence like (7).

(7) That the boy and the girl left amazed us.

The first step upon hearing the initial terminal *that* is to build a tree down from S of the form (8a).

(8) a.

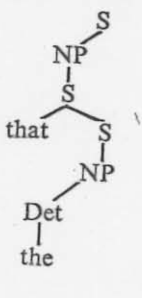


We may ask why such a tree is justified given that the initial terminal *that* could also have begun a sentence with a totally different structure, e.g. 'That is a nice flower'. To reduce the number of false starts, we may add the assumption that English is a look-ahead language. That is, the speaker is allowed to hear symbols subsequent to a given terminal before making up his mind as to the appropriate or most probable tree structure. I have no immediate empirical justification for this assumption, other than general considerations of simplicity and efficiency of parsing. The assumption is based on the conjecture that it may require less computation to hold a symbol in storage without tree attachment while one, but probably no more than two, subsequent symbols are scanned, than to build a tree only to have to return to alter it. (We will see later that sentences in which large tree changes are required during parsing are indeed perceptually complex.)

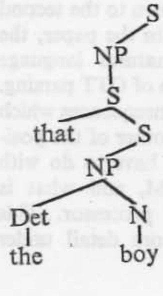
The reader will notice that complementizers are represented as Chomsky adjoined (*cf.* Kimball, 1972a, for a discussion of Chomsky adjunction) to the left of the appropriate notes, following Ross (1967). There is ample justification of this purely in terms of syntax. We will see later on that further justification arises from considering the role complementizers and other 'function' words play in perceptual routines.

Let us continue now to parse (7). Upon reading *the*, (8a) becomes (8b), which becomes (8c) after *boy* is read.

(8) b.

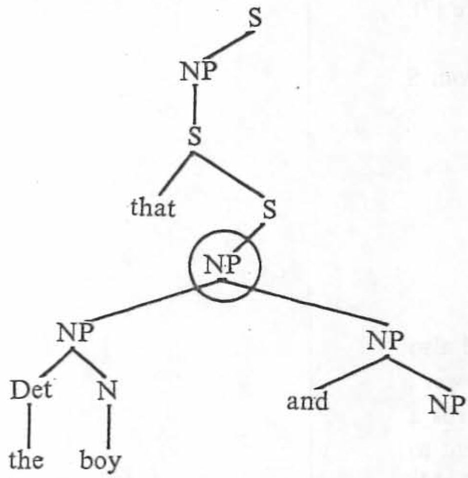


c.



At this point *and* is read, signalling a conjunction, and three new NP nodes are constructed, shown in (8d).

(8) d.



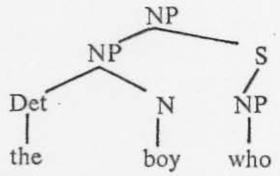
The *and* here is represented as Chomsky adjoined to the phrase following it. Again, justification for this can be found in Ross (1967). The circled NP is inserted between the lower S and the first NP. Insertion of such nodes I claim is possible, and such insertion is the only deviation from the general claim that trees are built top-down.[1] Furthermore, it seems to be the case that nodes that are so inserted are copies of already built nodes, so the S | NP structure is preserved.

Once the NP *the boy* is completed, all subsequent material is taken as belonging to a different phrase. If there were a relative clause construction, such as 'the boy who kissed the girl', as soon as *boy* is read, the NP is closed. Reading of *who* occasions building a new phrase, as shown in (9).

1. As this type of parsing differs from the usual top-down procedures, we may seek a new name for it. De Remer suggests over-the-top (OTT). In fact, it may be suggested that the mechanism of parsing in fact utilised in natural language is this: Trees are not built down to single terminals but with regards to adjacent pairs of terminals (discriminant pairs). Given an initial member of a pair, a tree is built over-the-top down to the second member. This could be done in one of at least three ways: (1) The tree is built up only as far as the lowest common dominating node for the pair under consideration; (2) the tree is built up only as far as the lowest common dominating S node for the pair, and then down to the second member; or (3) the tree is built all the way up to the highest S node, and down to the second member. As I have given it in the paper, the parsing hypothesized for natural language corresponds to this third type of OTT parsing.

There are some testable consequences which result from taking one or another of the positions outlined above. These have to do with what is maintained in STM, and what is cleared into the syntactic processor. This question is discussed in more detail under principle seven below.

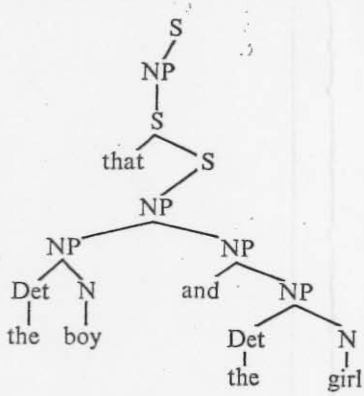
(9)



This observation concerning the closing a phrases is illustrative of a principle that operates generally in sentence parsing. This principle of closure will be discussed in detail below and is connected with another principle dealing with how semantic information is processed.

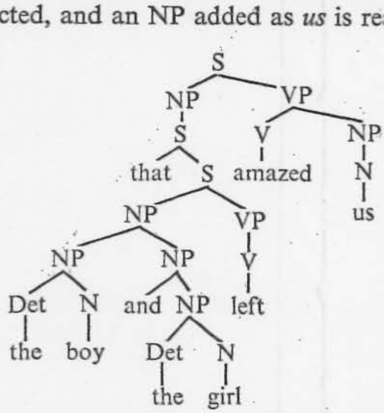
Returning once more to (7), after *the* and *girl* are read, the tree looks like (8e).

(8) e.



When *left* is read, the embedded sentence is closed, following the principle alluded to above, given look ahead to make sure no other possible parts of that sentence occurred (e.g., *early*, or *to NY*). When *amazed* is read a new VP in the main sentence is constructed, and an NP added as *us* is read. The final parse tree is shown in (8f).

(8) f.

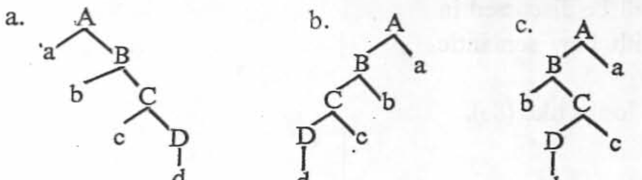


The 'top-down' principle interacts in an interesting way with the following principle.

4.2 *Principle Two (Right Association): Terminal symbols optimally associate to the lowest nonterminal node.*

This principle is designed in part to explain the frequently observed fact that sentences of natural language organize themselves generally into right-branching structures such as (10a), and that these structures are perceptually less complex than left-branching structures, such as (10b), or center embedded structures, such as (10c).

(10)



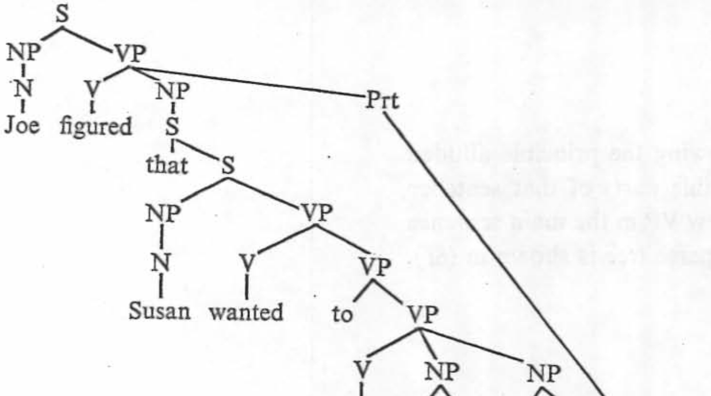
There is considerable evidence for the existence of such a principle.

First, consider a sentence such as (11).

(11) Joe figured that Susan wanted to take the train to New York out.

The surface structure of this sentence is shown in (12). It will be noticed that the particle *out* must be associated with a node other than the lowest (and, thus, rightmost). That is, 'out' is not associated with the node dominating 'New York' but with a higher node.

(12)



This principle also explains why a sentence that is potentially ambiguous, such as 'Joe figured that Susan wanted to take the cat out' is read by speakers naturally in a way such that 'take the cat out' is a phrase.

The status of such sentences constituted a puzzle for Bever (1970a) in that no known principle other than general 'memory limitation' would explain the difficulty of such sentences; he was unsure whether they should be marked ungrammatical or merely unacceptable. The sentence Bever gives is (37) (his numbering).

(37) I thought the request of the astronomer who was trying at the same time to count the constellations on his toes without taking his shoes off or looking at me over.

Such sentences should be counted as fully grammatical in that they are generated by general syntactic mechanisms. Their perceptual complexity is explained by Right Association, which is related to the principle of Closure discussed below.

Right Association accounts for the difficulty with phrases like 'the boy who Bill expected to leave's ball', or the preferred but incorrect interpretation of 'the boy who Sam gave the ball's book' (incorrect reading is that it's the ball's book; correct is that it is the boy's book). The reason is that the possessive *'s* optimally associates with the lowest constituent, instead of the higher NP dominating the whole phrase.

This principle also explains the preferred interpretation of a sentence like (13).

(13) The girl took the job that was attractive.

Which is that which is not synonymous with (14).

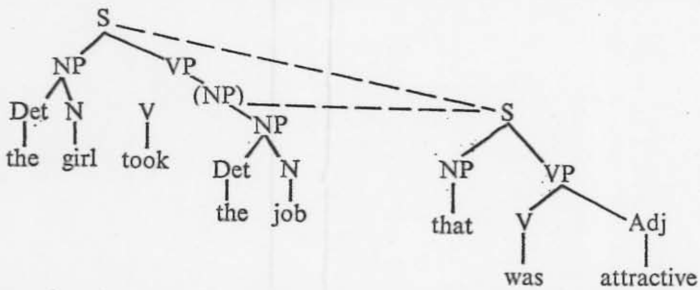
(14) The girl that was attractive took the job.

However, there is a general grammatical process that would form (13) out of (14), known as *Extraposition From NP*. This transformation maps a sentence like (15a) into (15b).

(15) a. The girl that was attractive went to NY.
b. The girl went to NY that was attractive.

The reason for the preferred interpretation of (13) can be seen from its parse tree (16).

(16)



Principle Two predicts that the relative clause will be perceived as associated with the lowest, rightmost node, i.e., *the job* rather than as a daughter of the top S, where it would have to be interpreted as an extraposed relative.

The seeming unambiguity of sentences like (13) have been taken as evidence for the existence of a special and probably quite powerful form of grammatical mechanism known as transderivational constraints. Briefly, according to the hypothesizers of such constraints, a derivation in which (13) comes from (14) is to be blocked because there already exists a sentence of the same form with a different interpretation. The fact is that (13) can be read as (14) with a little effort; it's just that this reading is perceptually more difficult, due to Principle Two. Thus, this datum should be removed as evidence for the existence of transderivational constraints. Without going into the matter in detail, I conjecture that all putative evidence for these devices are of the form of that above, namely, they can be explained in terms of preferred interpretation on the basis of established principles of perception. If so, there is no reason to include transderivation constraints among the stock of possible grammatical mechanisms in the theory of universal grammar. (But *cf.* Hankammer, 1973.)

There seem to be grammatical mechanisms to avoid the generation of sentences that would be perceptually complex under Principle Two, i.e., which would involve assigning terminals to other than the lowest, rightmost non-terminal. A transformation known as *Heavy NP Shift* is a case in point. This transformation moves heavy NP's to the right hand side of sentences, where definition of 'heavy' is discussed in Ross (1967).

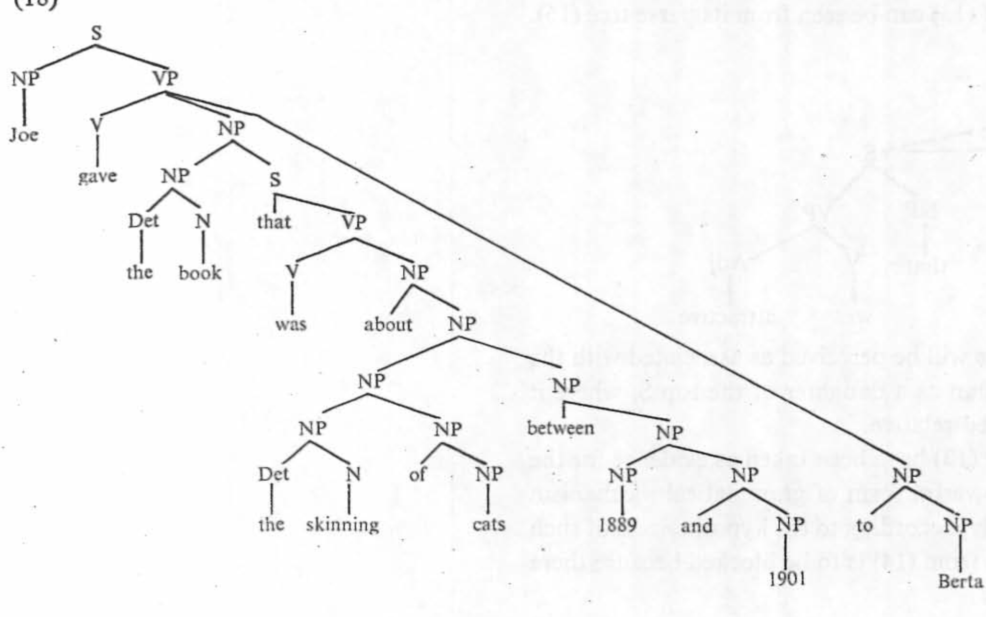
Thus, a sentence like (17a) would be mapped into (17b).

(17) a. Joe gave a book that was about the skinning of cats in Alberta between 1898 and 1901 to Berta.

b. Joe gave to Berta a book that was about the skinning of cats in Alberta between 1898 and 1901.

The perceptual complexity of (17a) can be seen in its surface parse tree (18).

(18)

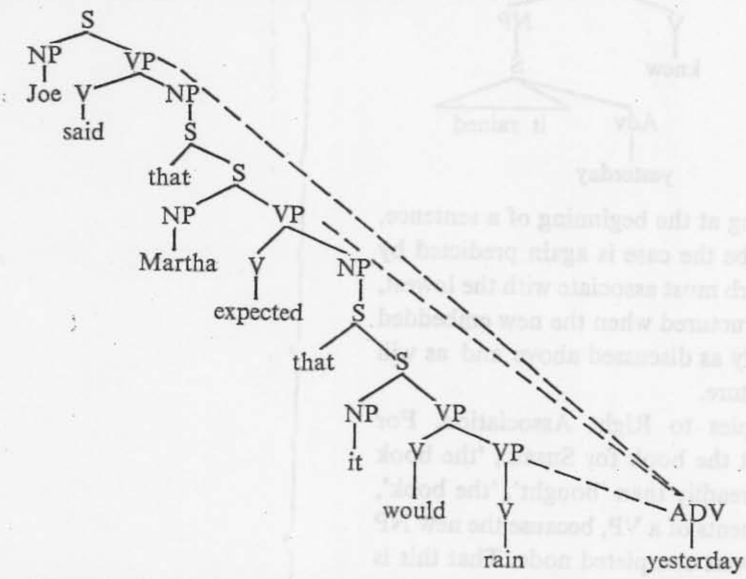


(It will be noticed that what are traditionally called prepositional phrases [e.g., *of cats*] are here represented as NPs with Chomsky adjoined prepositions. Justification for this may be found in Ross, 1967).

We can utilize the principle of Right Association in gaining a partial understanding of the complexity of sentences such as (1d) 'The girl the man the boy saw kissed left' which initiated the discussion of this paper. Part (but not all) of the difficulty resides in the fact that the verb *kissed* is optimally associated with the lowest, rightmost constituent of the tree. As this association is impossible on semantic grounds, it must receive association with a VP node in a higher sentence. The same goes for *left*. In this way (1d) violates Right Association.

Another confirmation of this principle comes from observing the natural association of adverbs. In this respect, consider a sentence like (19).

(19)



The dotted lines indicate the possible association of *yesterday*. Compare these with the natural associations of this adverb as the sentence is interpreted. The easiest reading is that in which the adverb is read as attached to the lowest VP, next easiest is that reading in which it hangs off the middle VP, and the hardest or least likely reading is that in which it is associated with *said.* This is exactly the prediction made by Right Association. In fact, we can define a metric such that the perceptual complexity of a sentence is proportional to the number of nodes above the lowest rightmost node to which a terminal is attached.

It is to be noted that there is a syntactic device available in English to disambiguate (19), as shown in (20a-c).

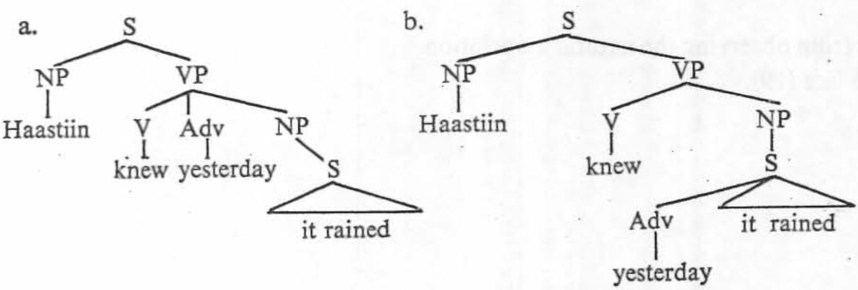
(20) a. Yesterday Joe said that Martha expected that it would rain.
b. Joe said that yesterday Martha expected it would rain.
c. Joe said that Martha expected that yesterday it would rain.

Notice that a sentence like (21) is read most naturally such that the adverb is associated with the higher clause.

(21) Haastiin knew yesterday it rained.

That is, the most natural structure imputed to (21) is that shown in (22a) rather than (22b),

(22)



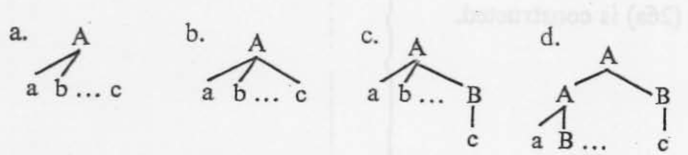
even though it is quite possible for an adverb to hang at the beginning of a sentence, as must be case, e.g., with (20a). That this should be the case is again predicted by the principle of Right Association. That is, the adverb must associate with the lowest, rightmost node. Conceivably, the tree could be restructured when the new embedded sentence is built, but such restructuring is very costly as discussed above and as will be elaborated below in the principle of Fixed Structure.

Let us consider some apparent counter-examples to Right Association. For example, this principle requires that in 'Joe bought the book for Susan', 'the book for Susan' should be interpreted as a phrase more readily than 'bought', 'the book', and 'for Susan' being interpreted as separate constituents of a VP, because the new NP 'for Susan' should preferably be assigned to the lowest completed node. That this is not the case is a function of the interaction of parsing with semantic information accessible to the speaker during the sentence scan. In particular, the verb 'buy' carries lexical information with it such that the speaker would expect to hear a 'for' phrase in its VP. This interaction of semantics with parsing will be discussed further below when the principle of *Processing* is presented.

In the same vein, notice that while 'Joe cooked the peas in the pot' is ambiguous, with either reading of equal complexity, 'Joe rode down the street in the car' does not carry the same ambiguity. That is, one does not read 'the street in the car' as a phrase because of semantics or knowledge about the world that streets usually aren't in cars. Again we see the role of outside information influencing parsing.

Thus, the above are not counter-examples to Right Association. Rather, this principle defines the optimal functioning of the parsing algorithm if no outside effects are relevant. Its operation may be superceded by mechanisms other than the parser.

The principle of Right Association operates in connection with another principle of perception, that of New Nodes. This principle is needed to account for the following observation: In processing a sentence, when the speaker has constructed a node A shown in (23a) and attached to it daughters a, b ..., upon reading the next terminal, *c*, Right Association demands that *c* be connected as shown in (23b). However, two other things may in fact happen. First, some new node, B, may appear and be subtended under A, as in (23c), or B may be Chomsky adjoined to the right of A, as in (23d).

(23)


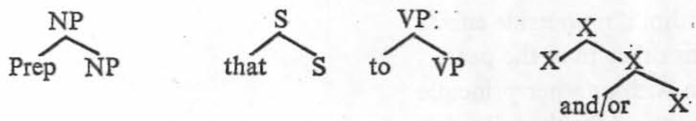
All three forms of assimilating the new terminal into the existing parse tree shown in (23b-d) are observed to occur in natural language. Right Association predicts that (23a) should become (23b). New Nodes is designed to predict when (23a) will become either (23c) or (23d), i.e., when the terminal *c* occasions the construction of a new phrase. This principle is stated as follows:

4.3 *Principle Three (New Nodes): The construction of a new node is signalled by the occurrence of a grammatical function word*

There is a traditional grammatical distinction in the discussion of the parts of speech between what are called content words (nouns, verbs, adjectives, etc.) and function words (prepositions, conjunctions, etc.). In the literature of transformational grammar, this distinction surfaces in terms of the difference between lexical formatives and grammatical formatives. For the time being I will focus on just prepositions, *wh*-words (e.g., *what, where, who, how, when, why*, etc.) conjunctions, and complementizers (*that, for-to*, and *pos-ing*). Later other categories will be examined as to whether they work like function words for purposes of perception.

There is syntactic evidence that grammatical formatives are Chomsky adjoined on surface structure (*cf.* Ross, 1967). (The assumption that this is the case is in fact not necessary to the correct operation of New Nodes, but I shall maintain the assumption in that which follows.) Thus, what is traditionally called a prepositional phrase is in fact a NP, as in (24a), and the complementizers and conjunctions appear on surface structure as in (24b,c).

(24) a. b. c.



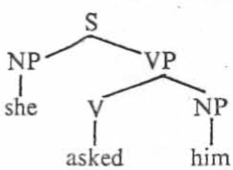
There is no direct proof that fronted *wh*-words, as in 'What did he say?' or 'The boy who you say' are Chomsky adjoined to the front of their clauses. As mentioned above, it makes no difference for New Nodes whether this is the case or not.

Let us examine how New Nodes operates to correctly predict the parsing of a sentence like (25).

(25) She asked him or she persuaded him to leave.

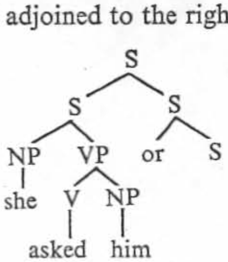
After reading the first three words of (25) a tree such as that in (26a) is constructed.

(26) a.



At this point a conjunction is reached, and the speaker must decide whether there is a conjunction of NPs (She asked him or her to leave), of VPs (She asked him or persuaded him to leave) or of Ss, as in (25). In this case a look ahead of one word reveals that the latter is the case, and (26b) is constructed, where the new node is Chomsky adjoined to the right of the top S.

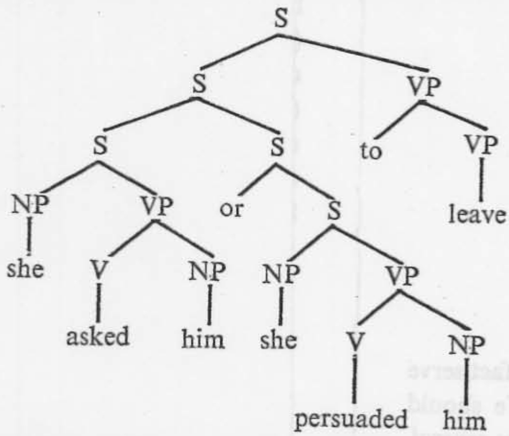
(26) b.



Right Association says that in this case the conjunction of NPs is easiest, as it is the lowest node, that of VPs next easy, and a conjunction of Ss hardest to perceive. This seems in fact to be the case for the sentences listed above. The deeper the node from which a conjunction proceeds, the more perceptually complex the sentence. An extreme case would be: 'Everyone said that Bill thought that Max believed that she was, although no one in his right mind who had been the movie would expect that Fred had told Sally that she was pregnant.' The perceptual complexity arises from the large internal constituent breaks, of which there are two in the above sentence, one before the conjunction *although*, and one before *pregnant*.

When the *to* of *to leave* in (25) is reached, this is a signal that a new node, in this case a VP, is to be formed. The structure of the conjunction forces this to be Chomsky adjoined to the top S, yielding a final parse tree (26c).

(26) c.



Thus, (25) illustrates how it is that both conjunctions and complementizers occasion the construction of a new node. New Nodes predicts that sentences in which the complementizers and relative pronouns have been deleted by optional rules will be perceptually more complex than those in which complementizers are present, i.e., that (27a) is more complex than (27b), and (28a) more complex than (28b).

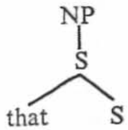
(27) a. He knew the girl left.
 b. He knew that the girl left.

(28) a. The boy who the girl who the man saw kissed left.
 b. The boy the girl the man saw kissed left.

There is experimental evidence to support this contention. Hakes (1972) found that sentences with complementizers were processed faster than those without complementizers. He writes: 'When an optional cue to a sentence's underlying grammatical relations is deleted, the difficulty of comprehending is increased. These results taken together with the numerous results on relative pronoun deletion suggest that the cue deletion effect is general and not limited to a particular cue or structure' (pp. 283-284). Thus, New Nodes supplies a second piece in fitting together an explanation of the difficulty of a sentence like (1d). (The third, and perhaps crucial, piece comes from Principle Four below.)

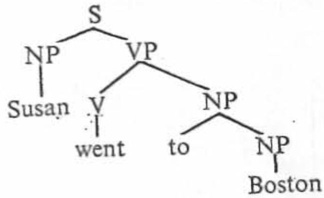
The particular example considered above illustrates only the nodes formed by a conjunction and the complementizer *to*. In sentences like (27a) or (27b), the complementizer *that* signals the existence of an embedded sentence. The occurrence of this complementizer here introduces a structure with three new nodes, as shown in (27c).

(27) a. Joe knew that it was a duck.
b. That it was a duck annoyed Joe.
c.



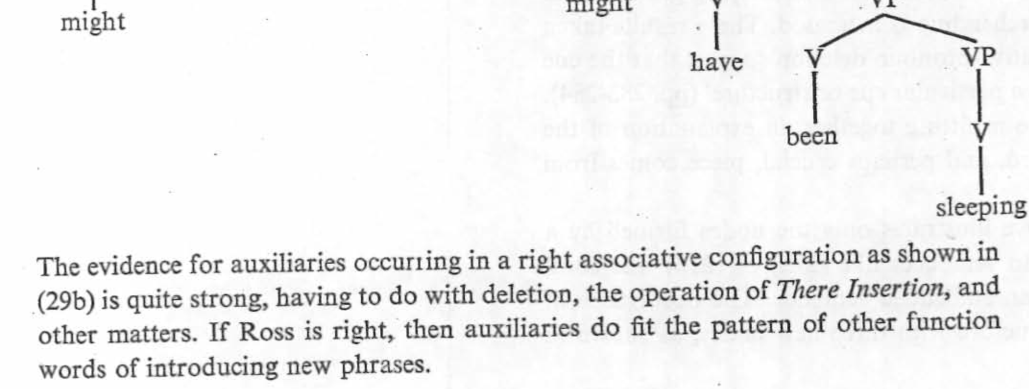
A preposition introduces the node NP, as in (28).

(28)



Traditionally articles (*a*, *the*) are included as function words, and they do in fact serve to introduce new phrases, although they are not Chomsky adjoined. We should include, perhaps, all words which fill the determiner slot in surface structure: *several*, *all*, *each*, *every*, *few*, etc. Finally, we may consider auxiliaries, which traditionally were counted as function words. There is a debate concerning the proper surface structure of the auxiliaries. Following the Chomsky (1957) analysis, it would be (29a); Ross' analysis (1967b) would predict (29b) as the correct structure.

(29) a. b.



The evidence for auxiliaries occurring in a right associative configuration as shown in (29b) is quite strong, having to do with deletion, the operation of *There Insertion*, and other matters. If Ross is right, then auxiliaries do fit the pattern of other function words of introducing new phrases.

This statement of the role of function words in sentence perception as signallers of new phrases includes no hypothesis concerning their semantic role or their syntactic origin in deep structure. Function words themselves are among the things learned later in the process of acquisition, the first stage being that of telegraphic speech (Brown and Bellugi, 1964). Likewise, they are generally absent from 'pidgins'. In both cases, the grammatical structures may be conjectured to be not sufficiently complicated (say, in terms of occurrence of embedded sentences) to require cues to surface structure. We may hypothesize that there is a certain permissible complexity of surface structures which do not require indicators of constituent organization. In a free word order language, not so much of the surface tree is relevant to a determination of the underlying syntactic relations, and surface structures in such languages may be flat and relatively uncomplicated compared with a language such as English; thus, such a language may not need overt cues to surface parsings.

The operation of New Nodes in SOV languages needs further examination. In such languages grammatical formatives typically follow those constitutents to which they are attached (as pointed out to me by Jorge Hankammer). For cases where the constituent is a simple NP with a post-position, the principle could be operative, as this NP could be stored until a look-ahead to the post-position gave clue to its syntactic status. For large constitutents such as S's with following complementizers, New Nodes simply is inoperative. Note, however, that such cases are not counter examples; New Nodes has the logical form of a conditional: If a grammatical function word occurs, it signals construction of a new phrase.

It is possible now to consider a principle which is pervasive in application, and which is the first principle directly bearing on what short-term syntactic memory limitations are.

4.4 *Principle Four (Two Sentences): The constituents of no more than two sentences can be parsed at the same time*

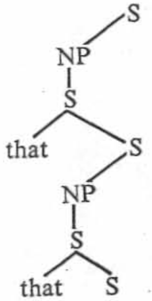
The first pieces of supporting evidence for this principle come simply from considering pairs of sentences like (30a,b) and (31a,b) with respect to complexity.

(30) a. That Joe left bothered Susan.
b. That that Joe left bothered Susan surprised Max.
c. That for Joe to leave bothers Susan surprised Max.

(31) a. The boy the girl kissed slept.
b. The boy the girl the man saw kissed slept.

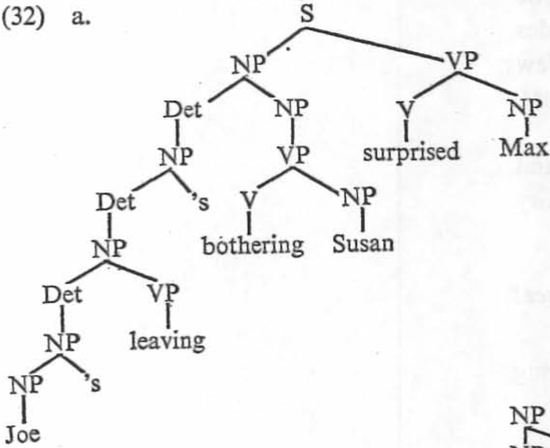
In processing both (30b, 31b) at some point the constituents of three different sentences must be held in memory. E.g., when the second *that* of (30b) is heard and recognized as a complementizer, the imputed structure is (30), where three unfinished sentences are being processed at once.

(30) d.

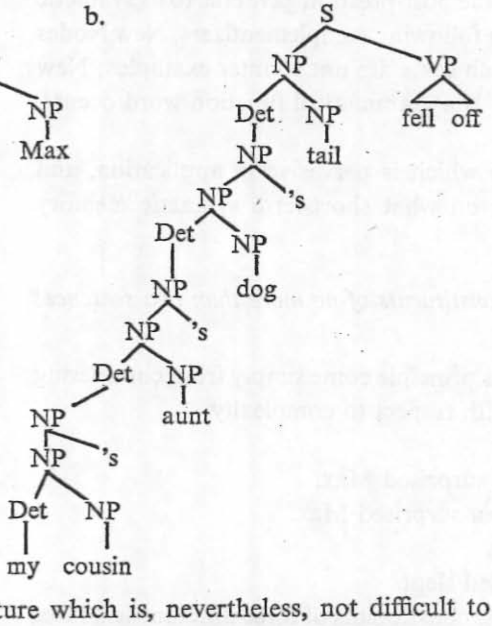


(30c) shows that repetition of 'that' does not here add noticeable complexity. Two Sentences provides the final principle explaining the difficulty of (1d). Part of the complexity of this sentence is due to the absence of *wh*-words to indicate the surface structure under the principle of New Nodes; part of the difficulty is that Right Association is violated; but the major difficulty here seems to be that the third sentence simply requires short-term memory space beyond the bounds of inherent capacity. Two Sentences is an attempt to state what that inherent capacity is. When the sentences in (30b) are nominatized, the result is much easier to parse, as in (32a).

(32) a.



b.



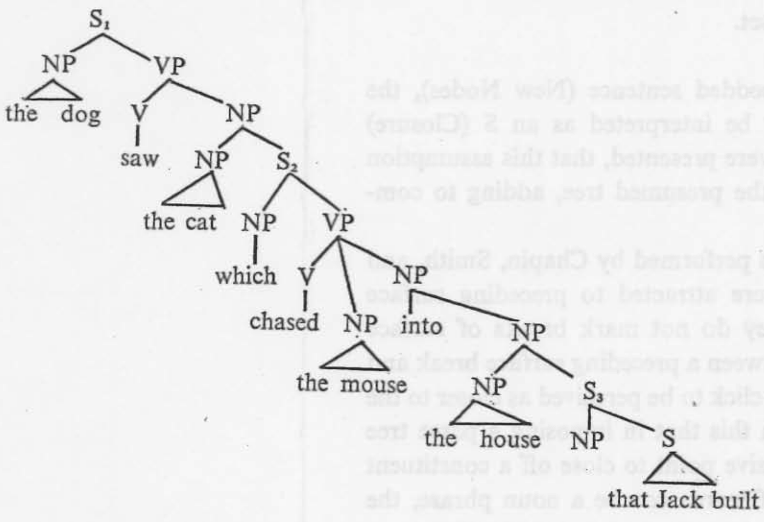
(32b) shows a large left-branching structure which is, nevertheless, not difficult to comprehend and which is within one S. On the other hand, when a fourth sentence is added to (30b in 33a), the result is totally incomprehensible, while the nominalized version, (33b), is not nearly as bad.

(33) a. That that that Joe left bothered Susan surprised Max annoyed no one.
b. Joe's leaving's bothering Susan's surprising Max annoyed no one.

One may conclude that the limitation is not on left branching, but rather on the number of Ss that must be processed at the same time. In a later discussion of semantic processing, I will discuss why this might be the case, and why it is permissible to string out relative clauses, embedded sentences, and prepositional phrases on the right of a sentence.

The Two Sentences principle accounts also for why right branching relative clause structures are permissible, while center embedded structures are not. Consider a structure like (34).

(34)



It may be thought that such a sentence violated Two Sentences because the top S is not finished until the last word of the bottom S is processed. To see why this is not so, let us say that we will consider a constituent to be 'closed' when the last immediately dominated rightmost daughter of that constituent is introduced in the process of parsing . In this sense, the top S is closed when the VP is reached, and the same for the second S. S_1 wouldn't be closed if, say, the sentential adverb *frankly* appeared sequentially after *built* and was to be attached to S_1. Why this definition of closure is appropriate and justified in that a phrase is through being parsed when it is closed will be discussed below under the principle of Processing. First, however, it is necessary to consider the principle of Closure.

4.5 *Principle Five (Closure): A phrase is closed as soon as possible, i.e., unless the next node parsed is an immediate constituent of that phrase*

Closure explains in part the complexity of sentences like (1c) 'The boat floated on the water sank'. In such sentences, as soon as the end of a potential S is reached, it is closed, unless the next phrase is also part of that S. Thus, at the end of 'the boat floated on the water', the assumption is that the S is closed. The remaining causes of the perceptual complexity of (1c) are accounted for below by Fixed Structure.

Also, the increased perceptual complexity of a sentence like (35b) over that of (35a) is explained by Closure, as well as New Nodes and Fixed Structure.

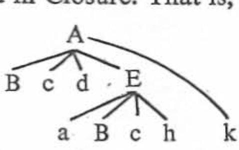
(35) a. They knew that the girl was in the closet.
b. They knew the girl was in the closet.

Without the complementizer to signal the embedded sentence (New Nodes), the sequence 'they knew the girl' would optimally be interpreted as an S (Closure) (modulo look-ahead), but when the later words were presented, that this assumption was incorrect would require a restructuring of the presumed tree, adding to complexity (Fixed Struture).

Evidence for Closure derives from experiments performed by Chapin, Smith, and Abrahamson (1972). They found that clicks were attracted to preceding surface structure constituent boundaries, even when they do not mark breaks of surface clauses. In particular, when a click was placed between a preceding surface break and a following clause break, the tendency was for the click to be perceived as closer to the preceding boundary. The authors conclude from this that in imposing a parse tree on a sentence, a subject 'attempts at each successive point to close off a constituent at the highest possible level. Thus, if a string of words *can* be a noun phrase, the subject assumes that it *is* a noun phrase and that the next element heard will be part of some subsequent constituent. Such a strategy would explain the strong preposing tendency observed in our experiment' (p. 171).

Closure has interacted closely with a number of the principles discussed above. In particular, it is not clear whether this principle is distinct from the principle of Right Association, for when the latter is violated, a terminal must be placed as a daughter of a node not the lowest, rightmost in the tree. Thus, this higher node must be 'reopened' to have a constituent added to it, contrary to the optimal situation described in Closure. That is, consider an abstract tree like (36).

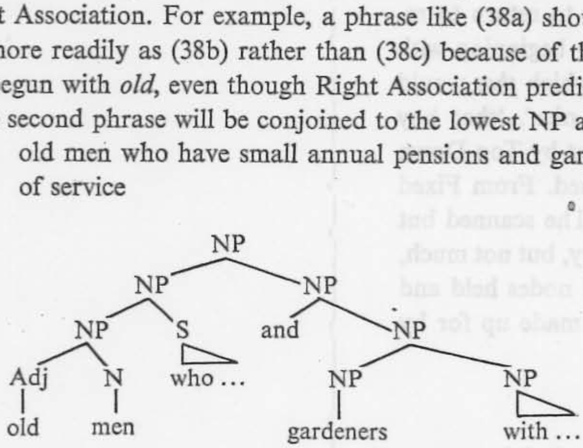
(36)



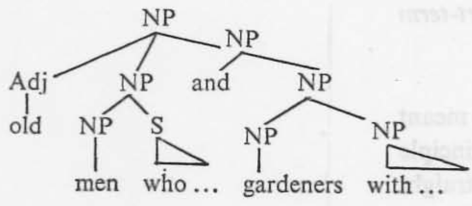
In building such a tree, as soon as E was being built, Closure would require that A be finished. However, when *k* is reached, A must be re-opened to receive the new constituent. In this sense, Closure operates the same as Right Association. However, I think that there are genuine cases where Closure operates which could not be covered by Right Association. For example, a phrase like (38a) should by Closure be interpreted more readily as (38b) rather than (38c) because of the tendency to close the phrase begun with *old*, even though Right Association predicts the opposite because by it the second phrase will be conjoined to the lowest NP available to it.

(38) a. old men who have small annual pensions and gardeners with thirty years of service

b.



c.



Bever (1970b) accounts for the difficulty in sentences like (1c) ('The boat Floated on the water sank') in terms of what he conjectures to be a general strategy of sentence perception. This principle (strategy B, p. 294) is that the first N ... V ... (N) sequence isolated in the course of parsing a sentence is interpreted as the main clause. This strategy is a particular case of Closure applied to sentences. Restated, it says that when an S node is 'opened' in the course of a parse, the first substring interpretable as an S (given some look ahead) will be so interpreted. In general, when a terminal string can be interpreted as an X-phrase, it will be.

With Closure established, we can now turn to Fixed Structure.

4.6 *Principle Six (Fixed Structure): When the last immediate constituent of a phrase has been formed, and the phrase E closed, it is costly in terms of perceptual complexity ever to have to go back to reorganize the constituents of that phrase*

This principle explains the complexity of sentences like (39a,b), as explained above.

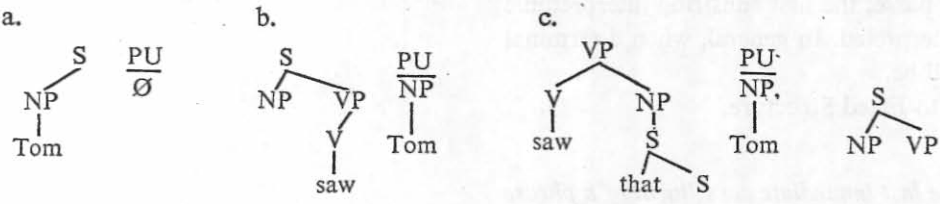
(39) a. The horse raced past the barn fell.
b. The dog knew the cat disappeared, was rescued.

The principle is connected with the look-ahead capacities of the sentence analyzer. Part of the function of this assumed capacity is to prevent having to return to reorganize previously assigned constituents. For example, a sentence beginning with *that* could be continued in at least three different ways, in each of which *that* would be the initial constituent of different phrases ('That 2+2=4 is nice', 'that boy sang', 'that is a big camel'). Thus, the initial tree built down to *that* by Top-Down will not be determined until succeeding terminals have been scanned. From Fixed Structure we can conclude that English *is* a look-ahead language. The scanned but unconnected terminals occupy a certain portion of short-term memory, but not much, in that the biggest restriction here seems to be on the number of S nodes held and being processed, and the allocation of storage space is more than made up for by efficiency of parsing.

4.7 *Principle Seven (Processing): When a phrase is closed, it is pushed down into a syntactic (possibly semantic) processing stage and cleared from short-term memory*

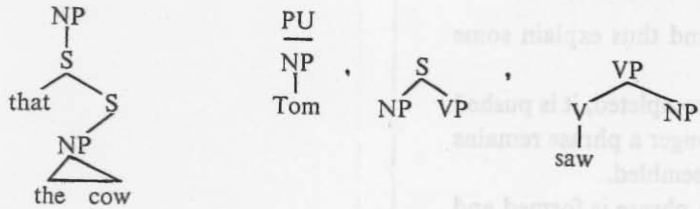
By this principle when a chunk of the tree is finished (where by 'a chunk' is meant a node and all its immediate constituents), it is sent to processing. This principle requires the assumption that there are pointers in the processing unit to keep straight the original structure of the tree, but such devices are simple mechanisms of data organization and surely can be documented to occur in other kinds of data processing (for example, association) that humans perform. Under this assumption, consider how a sentence like (40) might be processed. At each stage, the contents of the processing unit (PU) are listed.

(40) Tom saw that the cow jumped over the moon.

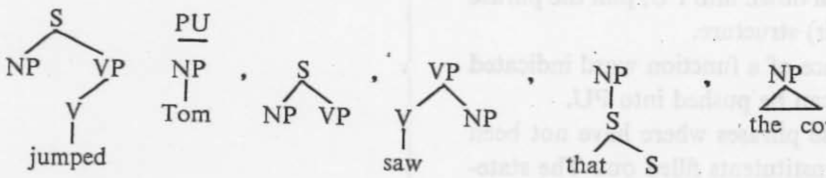


At this stage, e.g., there are pointers indicating that the first NP in the PU is that dominated by the S, and the VP dominated by the S is that currently still being worked on in short-term memory.

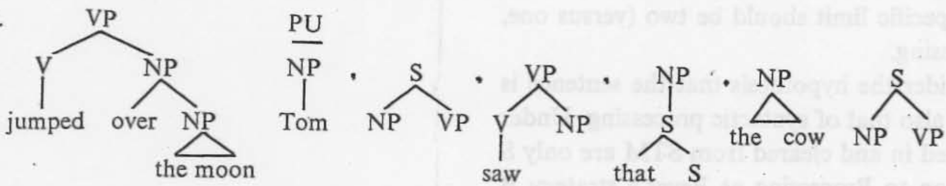
d.

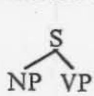


e.



f.



Again, pointers keep straight the relations among these tree chunks in the PU. E.g., there are two chunks of the form 

, and by the pointers it is kept straight which is matrix and which subordinate. Notice, in fact, that in a right branching structure the matrix sentence will always appear to the left in the PU of the embedded sentence. This suggests that the form of the tree on surface structure is relevant for the ease of its process in the PU.

I assume, further, that at any point semantic information in the PU is available for current decisions being made in constructing the parse tree, as in the different possibilities for parsing in 'They cooked the peas in the pot' absent in 'They rode the street in the car'.

Notice that at any given moment during the parse, not much more than a single phrase of one or at most two levels is held in short-term memory, a result of the fact that we have chosen for an example a sentence with a simple right branching structure. Sentences with center embedding require that a great deal more structure be held, simply because the higher phrases are not closed until the lower phrases are closed. Left branching structures don't present a problem, as each chunk of the tree is snipped off and placed in the PU as it is completed.

From the principle of Processing, it is possible to deduce and thus explain some of the principles discussed above, as follows:

(a) Closure results from the fact that as soon as a phrase is completed, it is pushed into the PU and thus removed from short-term memory. The longer a phrase remains uncompleted, the more of STM it takes up as its pieces are assembled.

(b) Fixed Structure follows from Processing because once a phrase is formed and pushed into the PU, then it should be difficult to reach down into PU, pull the phrase back out (plus all related phrases) to rework its (their) structure.

(c) New Nodes is explained because the occurrence of a function word indicated when a new phrase is begun and thus when the old can be pushed into PU.

(d) By Processing, what is held as STM are those phrases where have not been completed in the sense of having their immediate constitutents filled out. The statement of Two Sentences is thus made possible, for Processing establishes what is and is not in STM at any given time. That the specific limit should be two (versus one, three, four, etc.) does not follow from Processing.

In this connection it is interesting to consider the hypothesis that the sentence is not only the unit of semantic processing, but also that of syntactic processing. Under such an hypothesis, the units which are formed in and cleared from STM are only S units. This hypothesis bears the same relation to Processing as Bever's strategy B bears to Closure – namely, it is a specific case. At this time, I see no reason to prefer the restricted form over the mere general. I conjecture, then, that *syntactically* all phrases are treated alike in the process of establishing the surface tree. Semantically of course, S phrases occupy a special place. There is some evidence that the general Closure principle is correct – this same evidence seems to support Processing over the hypothesis considered above as Closure follows from Processing. If correct, this means that syntactically the unit of perception is the phrase – semantically the unit of perception is the sentence. This question deserves empirical investigation.

It is interesting to consider which among the seven principles adduced above are universal and which particular to English. I would conjecture that Processing and all those principles that follow from it deductively are universal, that it is common to all perceptual routines for parsing surface structures that phrases are formed, closed, and pushed into a processing unit, and that the semantic manipulations occur in that unit. Likewise, the condition on memory limitations stated in Two Sentences is probably universal. Thus, except for Top Down, which is an assumption for which it would be difficult to accrue evidence, it seems at first glance that all the principles above are universal. On the other hand, it would be most productive to look at a language like Japanese, which is 'backwards' from English in its order of constituents in the base, to determine which sentences are perceptually complex and why. Notice, again, that none of the principles predicts that left branching structures per se, as are found, e.g., in Japanese relative clauses, are difficult to parse. Center embeddings are difficult to process, as noted by everyone who has worked on the problem, and the principles above explain why.

Notice that the explanation of complexity of surface structures offered above does not refer to the transformational 'distance' of a surface structure from its deep structure. That is, these principles refer only to the tree pattern and not to how closely this resembles the tree pattern that represents the semantic relations in deep structure. In this way, these principles go against some of the earliest work on sentence recognition that sought to explain the complexity of a surface form in terms of its transformational history, which has also influenced later writings. For example, Foss and Cairns (1970) write: 'It seems reasonable to assume that the more the surface structure of a sentence distorts the grammatical relations in the structures underlying it, the more complex is the comprehension process and, hence, the more STS (short term storage) required to complete the task of understanding' (p. 541). If the principles above are correct, then distance between deep and surface structure bears no relation to sentence perception. It may bear on sentence comprehension and the nature of the computing process in PU, but this is a different matter.

So far I have said nothing concerning the internal structure of the PU, other than that its data input file consists of various tree chunks, plus an indication of how the pieces fit together. One model of the computations of the PU is that the surface tree is therein reformed, and transformations are applied 'backwards' to reconstruct a deep structure, which is then mapped into the meaning (under the identity mapping for a generative semanticist). I would conjecture that the sentence units of surface structure are reconstructed, being the basic units of comprehension. That is, in, say, 'John was believed by Bill to have been seen by Susan', part of the meaning is that John was believed and that John was seen.

5. Transformations and perceptual routines

Having a model of perceptual mechanism, it is now possible to discuss how various transformations and classes of transformations arrange the form of surface structure with respect to optimal perception. That is, one traditional explanation for the existence of transformations in natural language (Chomsky, 1965, chapter 1) is that they serve to arrange perceptually complex deep structures into perceptually simple surface structures. Given the definition of perceptual simplicity adduced above, it will now be possible to examine this claim in detail.

Research in transformational grammar has resulted in the accumulation of an inventory of transformations. The transformations which have been discovered, however, seem to fall into two distinct classes, the cyclic versus the last-cyclic transformations, with respect to a number of properties (Kimball, 1972c). That this should be the case is in no way predicted by the general theory of transformations. The properties of transformations in these two classes are listed below.

	Cyclic	*Last-cyclic*
(1)	Preserve form of input structure	Derange input structure
(2)	Make no essential use of variables	May make essential use of variables
(3)	May have lexical exceptions	No lexical exceptions
(4)	Several may apply within one S	Only one global transformation per S
(5)	Seem not to introduce structural ambiguities	May introduce structural ambiguities
(6)	Apply working upwards in tree	Apply only on top S

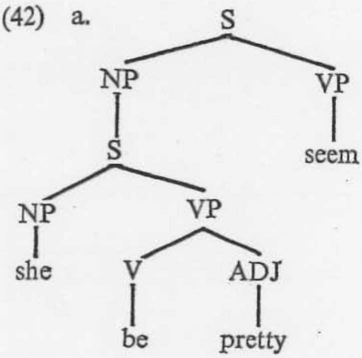
The last-cyclic transformations themselves can be divided into two groups, according to whether a transformation is global, moving constituents over a variable, or local. For example, a global transformation would be *Wh-Fronting* which moves question words to the front of sentences from arbitrarily far to the right. Thus, (41a) becomes (41b) by this transformation.

(41) a. He told you to ask Jill to go to the store to find *wh*-book?
b. *Wh*-book he told you to ask Jill to go to the store to find?
c. What book did he tell you to ask Jill to go to the store to find?

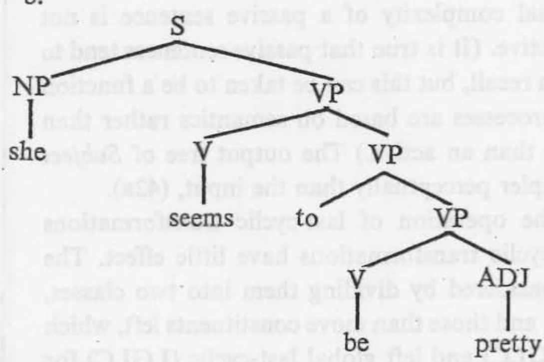
(41b) becomes (41c) by *Subject-Verb Inversion*, which is an example of a local last-cyclic transformation. As it turns out, the local transformations are essentially ordered with respect to some global transformations.

Perhaps the most interesting of the properties differentiating cyclic from last-cyclic transformations is the first, that cyclic transformations preserve the form of the input structures, while last-cyclic transformations tend to distort structure. For example, *Passive* operates on a structure of the form NP V NP, and produces an output structure of essentially the same form. *Dative* maps the structure V NP NP into one of the same form. An operation like *Equi NP Deletion* deletes an NP in an embedded sentence, but the sentence in the cycle on which it operates is unchanged. Likewise *Subject Raising* operating on a sentence embedded in subject position of a matrix sentence results in extraposing the VP of the embedded sentence to the end of the VP of the matrix, as shown in (42a,b).

(42) a.



b.

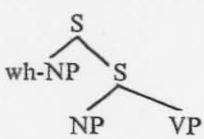


(The *to* is the remains of a *for-to* complementizer.)

Notice, however, that the form of the top S remains unchanged. Thus, the input structures to cyclic transformations are preserved across the operation of these transformations.

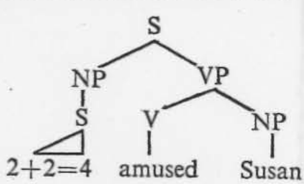
On the other hand, global as well as local last-cyclic transformations result in the production of structures unlike the input structures, structures which are also quite unlike those produced by the base rules of the grammar. For example, *Wh-Fronting* produces a structure shown in (43), leaving a hole where the *wh*-NP was extracted.

(43)

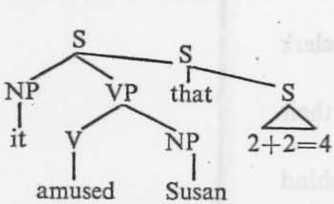


Extraposition moves sentences to the right, producing a structure like (44b) from (44a).

(44) a.



b.



It is interesting now to consider the effect of the transformations of these various classes on the tree with respect to the principles of surface structure parsing. The cyclic transformations generally leave behind a tree with the same or less complexity than the input tree. E.g., the perceptual complexity of a passive sentence is not discernably different from that of the active. (It is true that passive sentences tend to be remembered as actives over long-term recall, but this can be taken to be a function of the fact that these sort of memory processes are based on semantics rather than that a passive is more difficult to parse than an active.) The output tree of *Subject Raising* in subject position, (42b), is simpler perceptually than the input, (42a).

On the other hand, the effect of the operation of last-cyclic transformations requires some scrutiny. The local last-cyclic transformations have little effect. The global last-cyclic operations are best considered by dividing them into two classes, those that move constituents to the right, and those than move constituents left, which we may label right global last-cyclic (RGLC) and left global last-cyclic (LGLC) for convenience. Some RGLC's are listed below:

(45) a. *Extraposition from NP*
the boy who was tall left → the boy left who was tall

b. *Extraposition of PP*
a review of this book will be coming out → a review will be coming out of this book

c. *Extraposition* (44a) → (44b)

d. *Right Dislocation*
Joe gave the book that was about ducks to Susan → Joe gave it to Susan, the book that was about ducks

e. *Heavy NP Shift* (discussed in Section 4)
he asked the girl with the bright blouse to leave → he asked to leave the girl with the bright blouse

All of these transformations hang constituents out on right branches. They all may simplify the sentence in terms of principles like Right Association and Closure, for constituents internal to a sentence are made 'lighter', permitting closure to occur in general earlier. The relation of *Heavy NP Shift* to Right Association was discussed earlier. Note, however, that blind application of these rules does not in every case lead to a perceptually simpler sentence. E.g., (46a,b).

(46) a. He told the girl with the blonde eyelashes to go to the bank to ask the clerk to remove $100 from their account.

b. He told to go to the bank to ask the clerk to remove $100 from their account the girl with the blonde eyelashes.

Finally, notice that all RGLC transformations except *Heavy NP Shift* leave behind some place marker in the tree. That is, although some constituent is moved, a pronoun or some lexical material remains to mark the place of the removal, and this is generally not true of LGLCs. In terms of Processing, this does not add to the complexity. For by the time the extraposed material is reached and parsed, the original material will most likely be in PU, and a pointer can be attached to the extraposed constituent assigning it to that place in the PU.

Let us consider next some operations that move constituents to the left.

(47) a. *Topicalization* (moves an NP to the front of the main S)
Joe told Martha to ask Susan to test the bagel for Will → the bagel Joe told Martha to ask Susan to test for Will

b. *Wh-Fronting* (discussed above)

c. *Relative clause formation*
Joe spanked the child Bill had seen Betty kiss *wh*-child → Joe spanked the child which Bill had seen Betty kiss

d. *Left Dislocation* (Like *Topicalization*, except leaves a pronoun)
Joe gave the book to Sally → the book, Joe gave it to Sally.

Notice now that all these operations except (47d) leave no place marker to indicate the spot of removal. This may be an accident; however, from the point of view of the principles of perception none is required. How it is that the moved material is assigned the correct place for semantic analysis is a problem solved in the PU. One could imagine the moved constituents as being placed in a special category in PU awaiting the first possible pointer assignment to a spot in the tree as it is parsed. But the difficulty in discovering the surface constituents of the tree is not increased by these operations that move elements leftwards.

It could be conjectured that the RGLC rules leave a marker to indicate a place in the tree so that the extraposed material need not be assigned as a new constituent under some node already in PU. That is, with *Extraposition*, for example, a place is opened and held for the extraposed sentence by the *it*. Without some marker, to correctly appoint the encountered extraposed sentence to its place, a new NP would have to be entered under the S and a pointer assigned to that place. This would violate the principle of Fixed Structure, for some structure that had been placed in PU would have to be altered. Thus it is that it is possible to explain this property of RGLC rules with respect to the operation of rules of perception.

In summary, the cyclic transformations either effect no major change in structure from the point of view of perceptual complexity, or, in the case of *Subject Raising*, may operate to hang material on right branches. A right branching tree is not difficult to parse, as predicted by Right Association and Closure. As pointed out to me by DeRemer, a right associative structure is easier for a top-down mechanism which is predictive. On the other hand, left associative structure is much easier for a bottom-up parser. One may conjecture that languages such as Japanese with characteristic left branching employ a mixed strategy of bottom-up and top-down parsing. RGLC transformations hang material on right branches, simplifying the tree. The fact that these transformations leave markers behind in the tree to indicate the spot of removal is significant; an empty place remains in the tree for the extraposed material to be reassigned in the PU. Finally, LGLC transformations do not produce a tree that is perceptually more complex than the input structure. Their operation does require that the moved material be located back to the original place in the tree, but this is evidently performed in the PU, and so does not add to the *perceptual* complexity of the surface structure.

REFERENCES

Bever, T. G. (1970a) The influence of speech performance on linguistic structure. In G. B. F. d'Arcais and W. J. M. Levelt (Eds.) *Advances in psycholinguistics*. New York, American Elsevier.

Bever, T. G. (1970b) The cognitive basis for linguistic structures. In J. R. Hayes (Ed.) *Cognition and the development of language*. New York, John Wiley.

Brown, R. and Bellugi, U. (1964) Three processes in the child's acquisition of syntax. *Harv. educ. Rev.*, 34 (2), 133-151.

Chapin, P. G., Smith, T. S., and Abrahamson, A. A., (1972) Two factors in perceptual segmentation of speech. *J. verb. Learn. verb. Beh.*, 11, 164-173.

Chomsky, N. A. (1957) *Syntactic structures*. The Hague, Mouton.

Chomsky, N. A. (1965) *Aspects of the theory of syntax*. Cambridge, M.I.T. Press.

Chomsky, N. A. and Miller, G. (1963) Introduction to the formal analysis of natural languages. In R. D. Luce, R. R. Bush and E. Galanter (Eds.) *Handbook of mathematical psychology*, vol. 2. New York, Wiley.

De Remer, F. (1971) Simple LR(k) grammars. *Communications of the ACM*, 14, 453-460.

Fodor, J. A. and Garrett, M. (1967) Some syntactic determinants of sentential complexity. *Perc. Psychophy.*

Foss, D. J. and Cairns H. S., (1970) Som effects of memory limitation upon sentence comprehension and recall. *J. verb. Learn. verb. Beh.*, 9, 541-547.

Hakes, D. T. (1972) Effects of reducing complement constructions on sentence comprehension. *J. verb. Learn. verb. Beh.*, 11, 278-286.

Hankammer, J. (1973) Unacceptable ambiguity, *Ling. Inq.*, 4 (1).

Kimball, J. (1972a) *The formal theory of grammar*. New Jersey, Prentice-Hall.

Kimball, J. (1972b) The modality of conditions. In J. Kimball (Ed.) *Syntax and semantics*, vol. 1. New York, Seminar Press.

Kimball, J. (1972c) Cyclic and linear grammars. In J. Kimball (Ed.) *Syntax and semantics*, vol. 1. New York, Seminar Press.

Knuth, D. E. (1965) On the translation of languages from left to right. *Information and Control*, 8.

McKeeman, W. M., Horning, J. J., and Wortman, D. B. (1970) *A compiler generator*. New Jersey, Prentice Hall.

Ross, J. R. (1967) *Constraints on variables in syntax*. M.I.T. Dissertation.

Ross, J. R. (1967b) Auxiliaries as main verbs. Mimeo.

Résumé

Il existe dans la grammaire générative une distinction traditionnelle entre l'acceptabilité d'une phrase, qui appartient au domaine de la performance, et la grammaticalité d'une phrase, qui appartient au domaine de la compétence. Le but de l'article est de fournir une caractérisation de la notion de 'phrase acceptable' en anglais et de suggérer comment cette caractérisation pourrait avoir une portée universelle. La procédure consiste à donner une série de procédures qu'on pense être opérationnelles pour assigner l'arbre de la structure de surface à une phrase d'entrée. Ces principes d'analyse sont partiellement inspirés par les formules utilisées dans les langages d'informatique. Ces principes qui expliquent la grande acceptabilité de structures de dérivation à droite, mettent en évidence le rôle des mots grammatico/fonctionnels dans la perception des phrases, décrivent ce qui semble être une limite fixe de la mémoire à court terme dans le traitement linguistique et permettent de faire l'hypothèse sur la structure des mécanismes internes dans le traitement de la syntaxe. Enfin sont discutés les différentes classes de transformation que l'on peut utiliser pour préparer les structures profondes comme input à des procédures d'analyse.