

## The role of grammars in models of language use

ROBERT C. BERWICK\*

*Massachusetts Institute of Technology*

AMY S. WEINBERG

*Massachusetts Institute of Technology*

### Abstract

*This paper examines the question of whether and how the grammars proposed by linguists may be said to be 'realized' in adequate models of human sentence processing. We first review the assumptions guiding the so-called Derivational Theory of Complexity (DTC) experiments. Recall that the DTC experiments were taken to show that the theory of transformational grammar (TG) known as the Standard Theory was only a partially adequate model for human parsing. In particular, it was assumed (see Fodor et al., 1974) that the DTC experiments demonstrated that while the parser actually used the structural descriptions implicit in a transformational derivation, the computations it used bore little resemblance to the transformations proposed by a TG. The crucial assumptions behind the DTC were that (1) the processing model (or 'parser') performs operations in a linear, serial fashion; and (2) the parser incorporates a grammar written in more or less the same format as the competence grammar. If we assume strict seriality, then it also seems easier to embed an Extended Lexical Grammar, such as the model proposed in Bresnan (1978) (as opposed to a TG), into a parsing model. Therefore, this assumption plays an important role in Bresnan's critique of TG as an adequate part of a theory of language use. Both Fodor, Bever and Garrett (1974) and Bresnan (1978) attempt to make the grammatical rules compatible with the psycholinguistic data and with assumption (1) by proposing models that limit the amount of active computation performed*

---

\*We would like to thank Ed. Barton, Noam Chomsky and Norbert Hornstein for stimulating discussion and guidance during the writing of this paper. Joan Bresnan, Dick Carter, David Israel, William Marslen-Wilson, Curtis Marx, Ray Perrault, Lorraine Tyler, Tom Wasow, Patrick Winston, and three Cognition reviewers have helped improve the final version. Berwick is supported by the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's research is provided in part by the Office of Naval Research under Office of Naval Research contract N00014-80-C-0505. Reprint requests should be sent to R. Berwick, MIT NE43-820 or Amy Weinberg, MIT 20D 219, Cambridge Mass. 02139, U.S.A. The fault of any remaining errors, is, of course, in the stars. Note: Authors' names are in alphabetical order.

*on-line. They do this by eliminating the transformational component. However, we show that on-line computation need not be associated with added reaction time complexity. That is, we show that a parser that relates deep structure to surface structure by transformational rules (or, more accurately, by parsing rules tailored very closely after those of a transformational model) can be made to comport with the relevant psycholinguistic data, simply by varying assumption (1). In particular, we show that by embedding TG in a parallel computational architecture—an architecture that can be justified as a reasonable one for language use—one can capture the sentence processing complexity differences noted by DTC experimenter.*

*Assumption (2) is also relevant to the evaluation of competing grammars as theories of language use. First we show that Bresnan (1978) must relax this assumption in order to make Extended Lexical Grammar compatible with the psycholinguistic results. Secondly, we analyze Tyler and Marslen-Wilson's (1977) and Tyler's (1980) claim that their experiments show that one cannot instantiate a TG in a model of parsing without varying assumption (2). This is because they insist that their experiments support an 'interactive model' of parsing that, they believe, is incompatible with the 'Autonomy of Syntax' thesis. We show that the Autonomy Thesis bears no relation to their 'interactive model'. Therefore, adopting this model is no barrier to the direct incorporation of a TG in a parser.*

*Moreover, we show why meeting assumption (2), a condition that we dub the 'Type Transparency Hypothesis', is not an absolute criterion for judging the utility of a grammatical theory for the construction of a theory of parsing. We claim that the grammar need not be viewed as providing a parsing algorithm directly or transparently (assumption 2 above). Nevertheless, we insist that the theory of grammar figures centrally in the development of a model of language use even if Type Transparency is weakened in the ways that we suggest. Taken together, these considerations will be shown to bear on the comparative evaluation of candidate parsing models that incorporate transformational grammar, extended-lexical grammar, or the Tyler and Marslen-Wilson proposals.*

Generative linguists have insisted that the grammars they construct should be viewed as central components of psychological models of language use. One could reasonably interpret their claim as saying that the grammars proposed by linguists are somehow 'realized' in adequate models of parsing. This insistence is motivated by the reasonable assumption that a speaker/hearer should use the knowledge of his language (which linguists assume is described by linguistic theory) when processing or producing sentences.

It has also frequently been proposed that grammatical models be realized more or less *directly* as parsing algorithms. Evidently we are to impose the condition that the logical organization of rules and structures incorporated in a grammar be mirrored rather exactly in the organization of the parsing mechanism actually employed in sentence processing. We will call this the condition of *Type Transparency*. The Type Transparency Hypothesis makes a much stronger claim than one that holds simply that knowledge of language should guide the use of language: it claims that the principles employed to describe the system of knowledge that makes up the language faculty should also provide an adequate description of that system's implementation in language use.

There are clearly many ways that one could construe the Type Transparency Hypothesis. For instance, one could require an isomorphism between rules and operations of the *grammar* and the corresponding rules and operations of the *parser*. In its most literal interpretation this would mean that if a grammar proposes that sentence *X* is derived by using four transformations, then the parsing mechanism must take four operations in the analysis of *X*. This seems to be the position underlying work that falls under the rubric of the Derivational Theory of Complexity (henceforth DTC). This strict interpretation seems far too strong because weaker conditions are still compatible with the requirement of 'direct realization'. For instance, we might insist that the parser merely *preserve* distinctions made in the grammar (i.e., allow a homomorphic mapping); then the parser would be free to make additional distinctions. But the spirit of even this weakened condition still requires that more complex derivations in the grammar map over into more complex parsing operations in an order-preserving way; a derivation that takes five steps in the grammar should take, say, seven or eight steps for the parser. We could weaken the condition on homomorphism still further, as is done in Bresnan (1978). The mapping that Bresnan has in mind is one in which the condition of Type Transparency is not taken to be a relation between actual token rules of the grammar/parser pair. Rather, the distinctions between types of grammatical rules must be preserved as distinctions between types of parsing operations.

Bresnan begins by observing that the failure of the LTC has convinced many psychologists that "...no model of language use that incorporates a transformational grammar, or indeed any grammar, is reasonable" (Bresnan, 1978, page 2). By contrast, she claims that a psychologically realistic grammar should be such that "we should be able to define for it explicit realization mappings to psychological models of language use. These rules should map distinct grammatical rules and units into distinct processing operations and informational units in such a way that different rule types are

associated with different processing functions. If distinct grammatical rules were not distinguished in a psychological model under some realization mapping ... the grammar could not be said to represent the knowledge of the language user in any psychologically interesting sense" (Bresnan, 1978, page 3).

Bresnan claims that the value of this system lies in the fact that "theoretical linguistics has greatly advanced our understanding of the abstract structures of human languages. Under the conditions imposed, these advances could be brought directly to bear on the experimental investigation of human cognition" (Bresnan, 1978, p. 2).

Weakening the mapping between grammars and parsers still further, one might stipulate that the character of the levels of representation (e.g., Deep Structure, Surface Structure) must be preserved (either isomorphically or homomorphically) by the parser while the computational operations involved in mapping between these levels would be allowed to vary freely with respect to the rules postulated by linguists. In fact this is the position that Fodor, Bever, and Garrett argued that we are led to if we want to base our parser on a TG model.

Under any of these interpretations, one could still say that the parser operates so as to *interpret or generate sentences of L in the manner of G*, to use Chomsky and Miller's phrase (1963, page 399).<sup>1</sup>

This view contrasts rather sharply with that of Chomsky (1968) who in principle allows a much weaker connection between grammar and parsing algorithms:

...it is important to distinguish between the function and the properties of the perceptual model PM and the competence model G that it incorporates ... Although we may describe the grammar G as a system of processes and rules that apply in a certain order to relate sound and meaning, we are not entitled to take this as a description of the successive acts of a performance model (Chomsky, 1968, page 117).

Quite simply, Chomsky's position is that the grammar describes only *what* knowledge a speaker/hearer has of language; it does not prescribe any one particular parsing algorithm for how that knowledge is put to use. Note however that both the Type Transparency and the Chomskyan approaches take the grammar as at least *specifying the function to be computed by a parsing*

---

<sup>1</sup> Moreover, there are many positions intermediate between, on the one hand, absolute Type Transparency, and, on the other hand, a weak form of association between grammar and parser where the grammar specifies only the extension of the function that the parser computes. We will outline some of these alternatives in Section IV.



*algorithm*, that is, the grammar spells out which sentences input to the parser are to be considered members of a given language *L* and provides structural descriptions for these sentences.<sup>2</sup> One goal for a theory of parsing should be to investigate the range of algorithms (if any) that can compute this function.<sup>3</sup> On either view, the speaker/hearer's knowledge of language guides his use of language, and at the broader theoretical level, theories about the system of linguistic knowledge (the grammar) guide the construction of theories of parsing. This is so because both positions maintain that the theory of parsing is constrained to choose among algorithms that are capable of computing the function specified by the grammar.

The Type Transparency Hypothesis goes beyond Chomsky's position, however, in that it claims that the principles and rule systems involved in specifying the function mapping input strings to internal representations should also specify (perhaps completely) the actual procedure or algorithm used to compute that function. In one sense the intuitive appeal of this stronger view is easy to understand. The demand for a direct relationship between the theoretical objects of grammar and those of parsing would seem to allow experiments that tap into actual on-line processing to bear equally directly on the choice of both grammars and parsers for natural language.<sup>4</sup> For, if one can show that a principle assumed in a grammar *G* makes wrong predictions when incorporated in a parsing model, then, by the transparency condition, one can use this evidence to show that *both* the grammar and the

---

<sup>2</sup>The grammar furnishes multiple structural descriptions in the case of sentences that are structurally ambiguous, and (perhaps) incomplete or no descriptions for those sentences that are non-members of *L*. It even seems likely that a parser can successfully analyze ungrammatical sentences, so that structural descriptions of such sentences are available for further analysis so that judgments of grammaticality can be made.

<sup>3</sup>More carefully, these algorithms compute the *extension* of the function specified by the grammar. The requirement that human parsing procedures be algorithmic may be inappropriate. Supposing now that the human language faculty actually incorporates some sort of procedures that 'parse', it is not at all obvious just why those procedures must be algorithmic, in the formal sense of the term. For example, as Matthews (1979) has observed, one might be in full possession of knowledge of one's language (a grammar), and yet the parsing procedures might be unable to 'see' all the implications of the grammar's structure. As a case in point, consider Chomsky and Miller's (1963) well-known example of how a truncated-stack push-down automation can carry out the rules of a context-free grammar up to a certain point that depends on the depth of center-embedding of input sentences; the machine fails to carry out the rules of the grammar beyond this point. Such a device is in full possession of the production rules of the grammar (it 'knows' the language), yet it is unable to make full use of these rules in practice—it is not a complete decision procedure for recognizing the sentences generated by the grammar because it fails on some instances.

<sup>4</sup>The claim that the grammar-parser relationship might be exploited so as to bear on the theoretical choice of grammars has been expressed, for example, by Bresnan (1978, page 59): "But the grammatical realization problem can clarify and delimit the grammatical characterization problem. We can narrow the class of possible theoretical solutions by subjecting them to experimental psychological investigation as well as to linguistic investigation".

parser that incorporate this principle are inadequate. In short, invoking the Type Transparency Hypothesis in a strict fashion seems to guarantee that the 'external' measurements in the psychologist's current tool-kit will have much to say about the selection of the right theory of grammar, opening up a whole new domain of evidence bearing on the choice of an optimal linguistic theory.<sup>5</sup>

This would be a welcome state of affairs, if true: additional sources of evidence bearing on (underdetermined) scientific theories are always good news. It is crucial then to investigate whether the imposition of the Type Transparency Hypothesis can guarantee this promised connection between psycholinguistic observables and grammar.

In section I of this paper we begin our investigation by examining two proposals that have made crucial use of the Type Transparency Hypothesis. First we consider the Derivational Theory of Complexity (DTC), which assumes the Type Transparency Hypothesis. Next, we consider the Extended Lexical Theory as outlined in Bresnan (1978), a theory that was proposed in part to remedy the inadequacies of the DTC. Both positions take as a basic assumption the fact that *rules* (as opposed to *structures*) as they are stated in TG are not realizable in any but the weakest of the senses defined above.<sup>6</sup> Assuming (as discussed above) that the Type Transparency Hypothesis acts as an *a priori* methodological principle, sanctioning only direct mappings between grammars and parsers, it then follows that the legitimacy of transformational grammar as a description of linguistic competence is also undermined.

This argument has two flaws. First, in Section II of this paper we show that these conclusions follow only when the Transparency Hypothesis is conjoined with a *particular* view of human computational capacities. We will show that we can provide a model for the type transparent realization of a transformational grammar simply by embedding the grammar in an alternative parsing system. By an *alternative parsing system* we mean simply that one can posit other measures of computational complexity that can be embedded in a machine that incorporates these measures. (We outline one such proposal that employs the Parsifal parser of Marcus [1980]). Thus, invocation of the Type Transparency Hypothesis does *not* guarantee that psycholinguistic results can choose between competing grammars. In short,

---

<sup>5</sup>Miller and Chomsky (1963, page 471): "... the psychological plausibility of a transformational model of the language user would be strengthened, of course, if it could be shown that our performance on tasks requiring an appreciation of the structure of transformed sentences is some function of the nature, number, and complexity of the grammatical transformations involved."

<sup>6</sup>See Fodor, Bever and Garrett (1974, page 322) and Bresnan (1978, page 2).

*both* transformational grammar and extended lexical grammar can be made to meet a version of the Type Transparency Hypothesis, while retaining compatibility with the relevant psycholinguistic evidence.

More broadly, this result shows that the evaluation of psycholinguistic experiments is perhaps more complicated than has previously been thought. The proper evaluation of competing parsing procedures only makes sense if one can supply two things: (1) the procedures to be compared written in a uniform language (an algorithmic language); (2) an underlying theory of computational complexity, that is, a (possibly abstract) specification of a machine (its architecture plus explicit costs for each primitive operation of the machine), and how the procedures specified in the algorithmic language 'execute' on that machine. In using psycholinguistic experiments to choose between grammars it is *not* sufficient to present *one* parser (incorporating some grammar) that can perform a certain task. Rather, one must justify at least in a preliminary way both the grammar *and* the theory of human computational capacity underlying the parser. More particularly, in order to use psycholinguistic evidence to show that one grammar is more highly valued than another one must provide an *independently* plausible theory of computational capacity that yields the correct predictions for the experimental data most naturally when coupled with that particular theory of grammar. We will see that none of this has been shown. We conclude that current parsing evidence is *neutral* with respect to the choice between candidate grammars for natural languages.<sup>7</sup>

Secondly these arguments arguing against TG are flawed in their assumption that adequate grammars *must* meet the Type Transparency Hypothesis. We will claim that even if a transformational grammar could not meet this condition this should not be construed as a decisive argument against the grammar. As we will show in Sections III and IV, it is unwise to grant *a priori* methodological preference to theories that comport with the Type Transparency Hypothesis.

In Section III we deal with certain experimental work that graphically illustrates the danger of taking the Type Transparency Hypothesis too literally, that of Tyler and Marslen-Wilson (1977) and Tyler (1980). This work claims that psycholinguistic evidence shows that a model of language use incorporating the Type Transparency Hypothesis precludes the direct realization of a TG as a parser.

We argue that this claim is fallacious. The logic of Tyler and Marslen-Wilson's argument must assume that a grammar adhering to the autonomy of

---

<sup>7</sup>By 'candidate grammar' we mean a grammar that is otherwise explanatorily adequate, i.e., adequate for providing solutions to other psychological problems, such as the acquisition of language.

syntax thesis is compatible with one and only one derivational model (the model of non-interacting components usually associated with the *logical* organization of a TG). If one falsifies this derivational model then the autonomy of syntax thesis is likewise false. In other words, for Tyler and Marslen-Wilson's argument to go through, they must show that the autonomy of syntax thesis is compatible only with a non-interactive model. In contrast, we show that the autonomy of syntax thesis is consistent with many processing models, and in particular it is even consistent with the alternative parsing organization that Tyler and Marslen-Wilson propose. Again we will see that parsing evidence is neutral with respect to the choice between currently proposed grammars.

Section IV deals more directly with the notion of Type Transparency as a theoretical principle. In this section we outline how a theory that does not assume Type Transparency may still express the relationship between a theory of knowledge of language and a theory of language use. We will present a formal way of stating this relaxation of transparency *via* a device that has some currency in the study of parsers for programming languages, the notion of a *covering grammar*. Finally, we will suggest just why the relaxation of Type Transparency may lead to more fruitful avenues of research than a methodology that assumes the *a priori* reductionism inherent in the Type Transparency Hypothesis.

## 1. The Derivational Theory of Complexity

We turn first to theories that assume the Type Transparency Hypothesis. The classic (and perhaps simplest) view of a direct relationship between a theory of grammar and a theory of parsing is embodied in the so-called 'Derivational Theory of Complexity' [DTC]. First proposed by Chomsky and Miller (1963), it was later the subject of a flurry of psycholinguistic experimentation in the mid through late 60's. (This work is summarized in Fodor *et al.* [1974] and Levelt [1974].)

At the core of the DTC is a simple set of theses about parsing, one about *what* representation is constructed during a parse, and another about the *time course* of the parse itself. The DTC makes the following claims:

- (1) The so-called Standard Theory (the theory outlined in *Aspects of the Theory of Syntax*) is the optimal theory of grammar. By the Type Transparency Hypothesis, this means that sentences must be analyzed by a direct processing analogue of the Standard Theory. This analogue is specified as follows:
- (2) When one parses a sentence one recovers both the deep and surface structure representations of the input string of words. The surface structure is built up by

consulting the phrase structure rules of the grammar and matching them against the input string. The deep structure is derived from the surface structure by applying 'inverse transformations', if any such are specified by the transformational grammar to be involved in the mapping between the string's deep and surface structure; otherwise, the deep structure is just 'read off' the corresponding surface structure.

(3) The parser is organized so that each grammatical operation used to build either the surface structure or deep structure has a corresponding parser action that can be assigned a unit time cost. That is, in order to be counted as an 'active' component of the computation, each grammatical operation must take a unit of time to compute. A parser containing a grammar that maps between deep and surface structure by applying transformations would thus assign a unit cost to each one. Moreover, each such transformation is computed one at a time, i.e., serially. Thus the total cost of constructing the deep and surface structures is simply the sum of the total number of rules involved in the derivation of the sentence. Thus the relevant measure of complexity here is taken to be reflected in the *time* required to complete a parse.

Under the assumptions of the DTC a passive sentence would be expected to cost one more unit of processing time than an active sentence, because there would be an extra operation, the passive transformation, involved in the mapping between deep and surface structure for passive sentences.<sup>8</sup>

This hypothesis was investigated experimentally, and early work (experiments by McMahon, 1963 [Reference note 5]; Gough, 1965; Savin and Perchonock, 1965) seemed to support it. However, later investigation seemed to disconfirm the DTC, with the final coffin nail being supplied in the minds of many by Slobin (1966) and also Walker *et al.* (1968). Slobin presented subjects with pictures of action scenes that were described by either passive or active sentences. Subjects were asked to verify whether the supplied sentences truly described the corresponding pictures. Given the DTC, pictures described with passive sentences should be associated with longer verification times. This is because it was assumed that the task necessitated retrieving the deep structure level, and the passive sentences required one more operation than the active ones to effect this retrieval.

However, this expected difference only showed up in the verification of pictures described by *reversible* passive sentences, i.e., sentences such as *John was loved by Mary*, where either argument *John* or *Mary* may be reasonably interpreted as the Subject or Direct Object of the sentence. These

---

<sup>8</sup>Fodor, Bever and Garret, *The Psychology of Language*, page 229: "Further, the data suggest that the transformations may produce a linearly additive complication of the stimulus sentences: for example, sentences which involve both the negative and passive transformations appear to require a time approximately equal to the sum of the average time required for negative or passive applied separately."

sentences are to be contrasted with *non-reversible* passives, such as, *The cookies were smelled by John*. Here, only the animate *John* can be interpreted as the Subject of the sentence because *cookies* are incapable of smelling. The point of course is that since both reversible and non-reversible passives have the same *syntactic* structure, any account of the complexity difference between these two sentence types must be assigned to a different component of the grammar. (Slobin suggested that this cost differential be associated with a hypothesized semantic component.) These results disconfirm the DTC, particularly when one compares reversible passive sentences against active sentences. This is because according to the DTC *both* reversible and non-reversible passives should take longer to compute than their active counterparts, because both involve the same number of transformations from deep to surface structure. These results have been interpreted (see Fodor *et al.*, 1974) as showing that the transformational component could not be contributing to the computational complexity of sentence processing in these cases at all. Together with the assumption that all grammatical rules actively involved in sentence processing must have an associated unit cost, it follows that transformations are not 'realized' as *active* components of this kind of parsing model.<sup>9</sup>

It should be pointed out that the validity of Slobin's results has recently been questioned. Forster (1976 [Reference note 4]; 1978) has noted that:

Furthermore, using other experimental techniques (the RSVP presentation, Forster and Olbrei, 1973), reaction times for passives were found to be significantly longer than those for actives when subjects were asked to decide whether sentences were 'intelligible and grammatical'.

If Forster is right then the correct parsing theory for English should predict that passives are more complex than corresponding active sentences. By the Type Transparency Hypothesis this complexity distinction should be reflected in the grammar. Moreover, Forster and Olbrei's experiments showed *no* reversibility effects once sentences were controlled for another variable, plausibility.

We take no stand as to whether Slobin's results are valid or not. What interests us is the assumption that the validity of these results constitutes a barrier to the direct realization of a TG within a parsing model. We should be clear that our definition of 'direct realization' entails a parser that posits (1) levels of deep and surface structure (in the ST model) or annotated surface structure and S-structure (in the EST model), as well as (2) an analogue to the transformational component to map between these two levels.

---

<sup>9</sup>This was interpreted by some to mean that grammars could not be "psychologically real" See Chomsky [1980, pp. 189–197] for discussion showing the fallaciousness of this argument.

Thus our position is to be contrasted with that of Fodor *et al.* (1974) who claim that results like Slobin's show that "the grammar is probably not concretely realized in a perceptual model" [1974, pp. 369–370]. By this they mean that on-line sentence comprehension does not normally make use of a transformational component. They contend that the parsing functions previously attributed to transformations are in fact performed by 'heuristic strategies'. (See Fodor *et al.*, 1974, pp. 356 ff.) The purpose of the heuristic strategies is to reduce or eliminate the amount of on-line computation involved in sentence comprehension. This was done because it was felt that the extra computational effort involved in computing (or undoing) transformations made TG unsuitable as a computational model of human sentence processing; transformations would necessarily be correlated with the time complexity of sentence processing, and these complexities were not observed.<sup>10</sup>

In Section 1.1 we show that the assumption that extra computation must necessarily be associated with added time complexity in experimental tasks is also what underpins Bresnan's (1978) critique of TG and guides the design of a computational model associated with her alternative theory of grammar, lexical functional grammar. In Section 2 we show why this assumption need not hold. We do this by presenting a model that allows simultaneous computation (relaxing DTC thesis 3 above), while directly realizing a TG in the parsing mechanism.

<sup>10</sup> FBG recognized that their perceptual strategies make no contribution to a theory of language acquisition. This commits one to the (perhaps strange) view that the knowledge speakers have of their language (as specified by a grammar) is never used in language processing. Recognizing this possibility, FBG attempt to show that grammars are somehow involved in parsing. Their first suggestion is that the grammar could serve as a kind of 'backup routine' when parsing heuristics fail:

"That is, there exist some well formed sentences to which they [the heuristics reb/asw] will not assign the correct structural descriptions. Such sentences must be recognized by resort to 'brute force' ... problem-solving routines in which the grammar is concretely employed... The function of the grammar is to provide a library of information about the structure in a language and the function of some of the heuristics is to access the grammar" (Op. cit., pp. 370–371).

On this account there is still no sense in which the parsing mechanism is a reflection of the grammar rules. Rather, the grammar rules and parsing heuristics form two unrelated systems, an oddly redundant state of affairs. It would seem preferable to assume that we could eliminate one of these systems, that is, assume that there is one system that governs (albeit in an indirect way) both language use and language acquisition. FBG provide such an alternative when they suggest that the grammar provides the functions that the parsing algorithms must compute (see the Introduction and Section IV for a more complete discussion of this point):

"Recognition procedures can be constructed by a simple and general algorithm from grammars... The process of learning a (first) language involves internalizing the grammar and applying this algorithm to construct the corresponding recognition procedure" (op. cit., p. 371).

This alternative will be discussed in the main body of the text. The point of course is that an adequate psychological theory of language must contribute both to the theory of language learning and to the theory of language use.

### 1.1 *The Extended Lexical Theory*

We have seen that Slobin's refutation of DTC led to his rejection of transformational grammar as a 'realized' component of sentence processing. Bresnan (1978) has also apparently taken these results to mean that the DTC has been effectively refuted. However, rather than exploring alternative methods of computational organization, Bresnan has opted for modifying the grammar so that it is compatible with the parsing organization as sketched in DTC thesis #3 above.<sup>11</sup>

The Bresnan (1978) approach differs from transformational grammar in essentially two ways.<sup>12</sup> First, Bresnan claims that no so-called Noun Phrase movement transformations<sup>13</sup> are part of the grammar or a model of sentence processing. Rather, these transformational rules are reformulated as rules of *lexical-functional interpretation*. In Bresnan (1978) it is asserted that *one* way for these rules to be embedded in a parser is as precomputed templates rather than as 'active' computations. This allows Bresnan to embed the modified grammar (Extended Lexical Grammar, or ELG) in a parsing model organized along the lines of DTC assumption #3 above. She claims that the compatibility of this particular grammar-parser pair with results like Slobin's provides a strong reason for preferring the ELG theory of grammar to a transformational one.

In order to understand exactly how these claims about grammar interact with those about parsing, it will be necessary to first outline the kind of grammar that Bresnan envisions, and then sketch one way of realizing it in a parsing model.

The main difference between transformational grammar and Extended Lexical Grammar is the method by which these theories relate the thematic argument structure of predicates to surface syntactic structure. Consider the following three sentences:

<sup>11</sup> There are several other experiments that purport to falsify the DTC and undermine the possibility of directly realizing a TG in a parsing model. We treat them separately in Appendix 1 because we feel that these experiments suggest minor modifications in the DTC, but leave its major theses intact. They thus provide *no* barrier to the direct realization of a TG in a parsing model.

<sup>12</sup> We should stress that the approach presented in the Bresnan (1978) paper has been modified and formalized in Bresnan [1982, forthcoming] and Kaplan and Bresnan (1981). Although these more recent formulations differ in detail from the earlier work, these differences are not relevant to the issues discussed in this paper. The 1978 paper provides a clear account of the purported differences between transformational grammar and extended lexical grammar, particularly in regard to proposed models of human sentence processing. Therefore, we have chosen to focus on the 1978 work. For a discussion of the computational complexity properties of LFG (the successor of the ELT theory) as outlined in Kaplan and Bresnan (1981), see Berwick (1981a) and Berwick (1981b, Reference note).

<sup>13</sup> E.g., Passive, so-called "Raising" constructions, *There*-insertion, and the like.

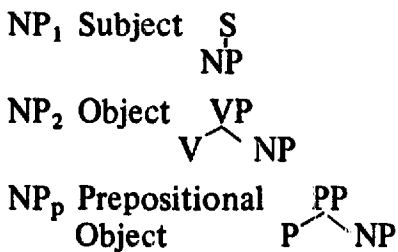


- (1) John sang the Messiah.
- (2) What did John sing?
- (3) The Messiah was sung by John.

In these sentences 'the *Messiah*' is interpreted as the patient of the verb *sing*. This fact is captured in transformational grammar by including a representational level of Deep Structure. In all three sentences, 'the *Messiah*' will be in the Direct Object position at this level of representation.<sup>14</sup> Transformations are then postulated that map the Deep Structure representation into surface syntactic structure.

ELT proposes to eliminate the Deep Structure level and the transformations for all the so-called Noun Phrase movement cases and derive thematic argument structure directly from the surface structure representation of a sentence.<sup>15</sup> This is done by "defining a set of lexical-functional structures that provide a direct mapping from the logical structure of a verb into its various syntactic contexts" (Bresnan, 1978, page 23).

Let us provide an example to make plain how this is done. In English, positions in the phrase structure tree are associated with certain *functional roles*. A functional role tells one the role that a given Noun Phrase plays in the interpretation of the sentence, e.g., whether a Noun Phrase is to be interpreted as the Subject, Direct Object, etc. For example, noun phrases in the following structural configurations (in English) receive the functional interpretations indicated below (from Bresnan, 1978, page 17):



These configurations capture the fact that in English the Noun Phrase directly dominated by S is interpreted as the Subject, the NP directly dominated by VP is interpreted as the Direct Object, and so forth.

Returning now to example sentence (1), the ELT theory maintains that the verb *sing* is entered into the lexicon with a functional representation like:

- (4) *sing*: NP<sub>1</sub> sing NP<sub>2</sub> (to NP<sub>3</sub>)

<sup>14</sup>In the Extended Standard Theory the Direct Object status of this phrase is also captured at the level of Surface Structure through the mechanism of *trace binding*.

<sup>15</sup>Transformations (or, rather, their interpretive counterparts) are retained for wh-movement *via* the device of binding.

This 'template' informs us that whatever NP fills the 'NP<sub>1</sub>' slot in the phrase structure tree will act as the Subject, whatever NP is in the 'NP<sub>2</sub>' slot will be the Direct Object, etc. The functional structure template is matched with the phrase structure tree that corresponds to sentence (1). *John* is interpreted as the Subject of this sentence because it is the NP dominated by S; likewise, *the Messiah* gets dubbed the functional Direct Object because it is the NP dominated by VP.

On the other hand, the mapping required to derive the correct thematic interpretation of e.g., the passive counterpart of (1) differs from the one needed for active sentences. The grammar must encode the fact that the position associated with the surface grammatical subject is athematic, and that the element in this position picks up the thematic role associated with the Direct Object position.

In the case at hand, the obvious rule way to do this is by a rule like the following (Bresnan, 1978, page 21):

Eliminate NP<sub>1</sub> ...  
Replace NP<sub>2</sub> by NP<sub>1</sub> ...

To encode the athematicity of the Subject position, Bresnan suggests that we bind a variable to this position.<sup>16</sup> The argument in this position in surface structure is then no longer associated with the thematic role normally given by the 'NP under S' position (it has been de-thematized) and so has no thematic role given to it. Thus following the rule above, this argument is associated with the thematic role of the NP<sub>2</sub> position.

So, *The Messiah be + past sung by John* is interpreted as ( $\exists x$  [*x be sung the Messiah by John.*]), informing us that in sentence (3) the Noun Phrase in the surface subject position is to be associated with the thematic role defined by (interpreted as) the functional Direct Object of this verb.

How is this modified grammar to be embodied in a parsing model? To ensure the most direct mapping, one could proceed in the following way: for the passive case we would need evidence from the input string that the 'typical' interpretation is to be blocked. This trigger can be supplied by the lexical entry associated with the form *sung*. Then, we would force the right non-canonical interpretation *via* the application of the lexical relation cited above. This would mean in the case just mentioned that given the phrase structure tree for the passive sentence above, we interpret the corresponding

<sup>16</sup>"But there is another way to establish a correspondence between the argument structure of a verb and its syntactic context... For example, the argument structure of *eat* can be converted from a two-place relation into a one-place relation. A logical operation that has precisely this effect is the variable binding operation of quantification" (Bresnan, 1978, p. 16).

sentence (3) by locating the NP directly dominated by S and removing any functional arguments associated with this position. Then the element in the NP<sub>1</sub> position would actually be placed in the Direct Object position (the NP<sub>2</sub> position), and would be thematically interpreted from that position.

But the theory-grammar pair as presented so far still does not square the ELT grammar with psycholinguistic results like Slobin's. (Still assuming now that the lexical redundancy rule mentioned above is an active computation performed by the parser.) This is because the interpretation of passive sentences costs one unit of time more than their active counterparts—the difference being exactly the processing cost of the lexical redundancy rule—and thus one is still left with a model equivalent to the DTC transformational version with respect to Slobin's timing results.

In order to make the model compatible with Slobin's results one could assume that the interpretation of the passive sentences is effected by comparing the surface string to a functional structure template that is listed in the lexicon as part of the entry of the corresponding verb just as in the active cases. (This is in fact the method that Bresnan (1978) suggests.) The *effect* of the lexical redundancy rule (rather than the rule itself) is encoded into the form of the functional template associated with a passive verb. An example may help make this clear. The functional template for the passive verb form of *sing* is:

Be + sing:  
( $\exists y$  [*y sing* NP<sub>1</sub> (to NP<sub>3</sub>)])

As this sentence is parsed the same matching operation would be effected as in the active case, but now the passive lexical form would be retrieved and the Noun Phrase in the structural subject position would be placed in the functional object position (as dictated by the template). This Noun Phrase would then be interpreted as the Direct Object, as desired. Since the same matching operation is involved in both the active and passive sentence—namely, the retrieval of lexical templates—then, given the additional assumptions of the processing model above, the processing of active and passive sentences will now take the same amount of time.<sup>17</sup>

In this model, the complexity distinction between reversible and non-reversible passives is *not* due to the relative complexity of retrieving the lexical templates. Rather, it is suggested that in all passives it basically is more difficult to give Noun Phrases in phrase structure trees their proper functional interpretations. This is because one cannot provide a *direct* assignment between NPs in the phrase structure tree and argument positions in

<sup>17</sup>Note that for this prediction to go through we must also assume that lexical retrieval takes "unit" time no matter how complex the passive form entry is relative to the active form entry.

functional structure; some kind of additional manipulation is demanded. Therefore, any extra cues that indicate which NP matches with which argument position can potentially speed up comprehension. Non-reversible passives contain such cues by virtue of the fact that the verb's selectional restrictions (e.g., whether it takes an animate or inanimate subject) admit only a single well-formed mapping between NP positions in the phrase structure tree and functional structural positions. For example, in a sentence like *The cookies were smelled by John*, once the verb *smell* is recognized, its selectional restrictions become available to the parser. The parser can immediately conclude that the NP<sub>1</sub> in the phrase structure tree cannot be the NP<sub>1</sub> of the functional structure, because only animate NPs may appear first in the functional structure associated with *smell*.

Finally it should be noted that in the extended lexical grammar both actives and passives contrast with, e.g., *wh*-movement constructions in that they are derived by lexical rules rather than by the transformations used to derive sentences with *wh* movements. Therefore, the recognition of actives and passives need not involve the same computations required to analyze *wh* constructions. Thus Bresnan assumes a weaker version of the Type Transparency Hypothesis: there is no one-to-one grammar-parser rule correspondence; there is, however, a *type* of grammatical rule—*type* of parsing computation correspondence. Since Bresnan (1978) treats *wh*-movement and passives separately in the *grammar*, she is permitted to assign these two rules to different *processing* components. In contrast, it is argued by implication that since transformational grammar retains both these rules as transformations, it must parse them by the *same* type of algorithm. Therefore it is claimed that ELT can capture similarities (between actives and passives) and differences (between passives and *wh* movement) that TG cannot.

It is important to point out that this argument depends upon a *particular* choice of computational organization. Given that the assumed parsing organization is so crucial to the argument against TG, it is important to investigate whether the same conclusions hold under alternative assumptions about computational organization. In the next section we will show that by allowing a rudimentary kind of parallel computation we can bring a transformationally-based parser into line with existing psycholinguistic reaction-time results. In particular we will claim that the passive morphology of the verb can act as a local cue, telling the parser that a certain *computation* must take place. (The computation is either *movement* or *binding*, depending on whether the parser is based on the so-called Standard Theory or Extended Standard Theory.) This computation can be effected concurrently with the recognition and attachment of the verbal element, and thus need not require any additional (externally measured) reaction-time.

## 2. Alternative parsing models

### 2.1 *Alternative interpretations of the DTC*

Let us assume that psycholinguistic results like Slobin's have in fact shown us that the DTC must be somehow modified. Recall that the DTC consists of three central theses, one about the type of grammar to be embodied as a parser, the second a strong version of Type Transparency (i.e., that either rule types or individual rules themselves are mapped directly to distinct parsing operations), and the third about the computational organization of parsing itself. To bring the DTC into line with experimental results, clearly we could modify any or all of these three assumptions. Bresnan (1978) has decided to alter only the first, adopting a different theory of grammar in lieu of a transformational account.

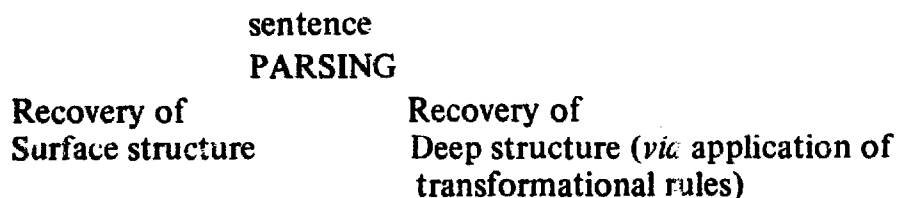
In this section we will see that by holding the grammar constant and varying the other two 'parameters' of the DTC, we can just as readily accommodate Slobin's psycholinguistic evidence. First we consider modifications to the computational organization of parsing. Our basic approach will be to introduce a slight amount of non-seriality (concurrent processing) into the execution schedule of parsing rules. We will show how the crucial non-concurrent processing can be triggered in the 'passive cases' upon recognition of the predicate of a passive sentence and how this triggering can be reasonably integrated into the machine architecture we have in mind.

We shall illustrate the impact of this non-serial processing by exhibiting parsing models for two transformational theories, the *Standard Theory* (see *Aspects of the Theory of Syntax* [Chomsky, 1965]), and the newer *Extended Standard Theory* (see Chomsky [1976]). Thus modified, both models will prove to be compatible with the DTC timing results.

### 2.2 *A parsing model for the Standard Theory*

For the purposes of constructing a parsing model, we need only the briefest review of the key premises of the Standard Theory (ST). Crucial to ST is the assumption that there is one level of linguistic representation relevant to phonetic interpretation, and one to semantic interpretation. Phonetic interpretation is 'read off' the surface structure of a sentence, while semantic interpretation is determined by the deep structure configuration. However, even though the theory specifies that two representations must be recovered from the input string, it does not specify in what order they must be recovered. Deep structure may be 'computed' after the entire surface structure tree is built, or, more to the point here, it may be built in parallel with

the on-going construction of the surface structure tree. It is this latter alternative that we shall adopt here. Thus the sketch we have in mind for the computational organization of parsing is the following:



In the following discussion we will incorporate this idea in a parser designed by Marcus (1980); the reader is referred to that source for details. We adopt this parser merely to be concrete; the concurrency scheme to be sketched is compatible with any number of parsing models. Further, the discussion in this section will require only an informal characterization of the parser we have in mind. For the purpose of understanding the discussion below, all that is important is that a Marcus-style parser operates by making decisions based upon two sources of information: (1) An ability to examine features of (i) the parse tree node currently under construction plus (ii) the features of a Noun Phrase or Sentence Phrase (*cyclic* node) immediately higher in the parse tree; this information is clearly useful in determining what the parse tree already built looks like, and hence what should be built next. (2) An ability to look at features of items in the input stream not yet attached to the parse tree, up to a limit, almost always, of three items (though this last constraint can be relaxed in some circumstances to admit a 'look-ahead' of five items). Together, predicates defined over (1) and (2) are used to determine the parser's next move. Clearly, the evidence the parser uses is of a strictly *local*, though abstract, sort, amounting to the examination of the features of nodes and input tokens in the 'immediate vicinity' of the parser's activity.<sup>18</sup> At any given step in a parse the Marcus parser can access the contents of five 'cells' in order to decide what to do next—two for the nodes corresponding to partially or completely analyzed phrases that will become part of the parse tree and three for the look-ahead. Crucially, *we will assume that access to each of these five cells takes only constant time*. Thus, the contents of these cells may be retrieved, examined, or modified, all in constant time.<sup>19</sup> Finally, we should stress that we are not interested in

<sup>18</sup>Moreover, since the number of features that a node like a Noun Phrase might have is by assumption finite, the total amount of information the parser has available at any given step in order to decide what to do next is also finite.

<sup>19</sup>Note that whether this computational ability is *in fact* available for human sentence processing is an empirical matter, and indeed the difficulty in verifying this (or any) specific computational organization is one of the major problems that we feel severely weakens the explanatory power of the Transparency Hypothesis.

justifying all the details of the Marcus parser; rather, we are interested in showing that by assuming *different* machine architectures we can radically alter what a particular grammar-parser pair predicts vis-à-vis reaction time and other measures of complexity.

Parsing an active sentence in this model is straightforward. Words enter the input stream (three at a time under most circumstances).<sup>20</sup> The surface structure and deep structure trees are built in parallel, elements in the input stream being placed simultaneously in their proper positions in each one. First the Subject Noun Phrase is assembled and then it is attached to the S node. Next, the Verb Phrase is assembled and attached. A sentence like,

(4) The girl kissed the boy.

would be parsed in roughly six steps of the parser: two steps to assemble the NP and one to attach it to the S, and two to assemble the VP and one more to attach it to the S.<sup>21</sup> In a simple active sentence, the deep structure tree will be isomorphic with the surface structure tree. Moreover, let us assume that recognition of the predicate of a sentence also entails the retrieval of its subcategorization and selectional frames (see Marcus [1980] for one way that this proposal may be carried out in detail). For our purposes we may assume that this information is available at any level (but see Chomsky [1981] for arguments that it must be available at *all* levels). Support for this assumption comes from the fact that experiments thought to tap on-line processing reveal significant complexity differences between the comprehension of 'anomalous' and fully well-formed sentences. In many cases, these two classes of sentences are distinguished solely by the property that the 'anomalous' cases fail to meet a predicate's thematic or selectional restrictions.

We now turn to a mechanism for analyzing passive sentences. Recall that the approach in Bresnan (1978) claims that the analysis of passive and active structures both involve lexical lookup. The fact that complexity increases are

<sup>20</sup> The left-right seriality of the order in which words enter the input stream is what allows us to mimic the left-right seriality of speech. It should be noted however that other components of the parser need not operate in this serial manner (though they certainly must wait upon the process that reads words into the input buffer); that is, in principle the machine computations on the input string may be effected in several different ways (for example, rules can operate in parallel over the items in the input buffer).

<sup>21</sup> The story is actually a little more complicated than this because of the way in which Parsifal actually assembles complete phrases. To construct a phrase, the parser first creates a category label in its active memory (like an 'NP'). Then it attaches elements to that category (like the Determiner and Head Noun), until the entire phrase is built. Next it drops the entire category (now with attached sub-constituents) into the input stream. Finally, it re-attaches the completely assembled phrase to the tree. We have shortened the exposition in the text because these details do not affect the relative complexities of the relevant examples.

found only in reversible passives is taken to be a function of the greater complexity in assigning the proper functional roles to NPs in the phrase structure representation of a sentence. This complexity is compensated for in non-reversible passives by the extra cues that non-reversibility provides to guide the NP-functional role mapping.

We will adapt this approach to a computational model based on TG. In particular, by using a modest amount of parallel processing (whereby two parsing actions can take place simultaneously) we will show that the analysis of passive structures (in particular, the recognition that there is an unfilled postverbal position that must be filled by an argument from another position in the surface string) takes the same amount of time as the recognition of corresponding active structures. This model can then be easily modified to incorporate devices that make the process of finding the proper NP more difficult than in the active case (the resolution is by actual movement in the Standard Theory or binding in EST). As in the Bresnan (1978) model, we will make use of selectional restrictions to drive a complexity wedge between the reversible and non-reversible passives.

Recall then that in the Marcus parser items in the input stream can be seen three to five at a time, by reading them into an input buffer. In almost all cases, this will allow parser rules simultaneous access to the Subject, Auxiliary verb, and verbal material of the predicate. To take a concrete example, consider the following sentences:

- (5) The eggplant was kissed.
- (6) The boy was kissed.
- (7) The girl kissed Fred.

During the parse of the first sentence, the parser's input buffer will first be filled with items as follows:

|the|eggplant| was |kissed|

Recall also that we have assumed that recognition of the verb entails recognition of its subcategorization and selectional frames. This approach assumes that nouns can be interpreted in terms of their features by the Marcus parser (although leaving open the question of just how these features are to be represented).<sup>22</sup> In this case, the actual representation in the input buffer would be:

|the|eggplant — Animate| was |kiss + ed| [Subject +  
animate \_\_\_\_ Object] | [ \_\_\_\_ NP]

<sup>22</sup>This is in fact what Marcus assumed.



There is a mismatch between the selectional demand of the verb for a +*Animate* Subject and the -*Animate* features of *eggplant*. Assume that this mismatch signals to the parser that the first NP is an unlikely candidate for the Subject position at Deep Structure. Thus, although the NP *the eggplant* can be built and attached to the Subject position of the *Surface Structure* tree that the parser is constructing, we can assume that a copy is left in the input stream for subsequent movement to a position that allows elements with its particular selectional features. So far then the construction of the *Surface Structure* tree will have required one more step than the construction of the corresponding Deep Structure tree (namely, the attachment of the Subject NP to the S node).

Next, just as in the active case, the parser assembles the VP by joining Aux and V nodes together. Crucially however, we assume that as it builds the VP it can recognize from morphological evidence (the *be*-V + *en* pattern) that the predicate is a passive participle. The parser then simultaneously (1) labels the predicate [+passive] and (2) attaches the V to the VP node under construction. Next, the passive morphology (as encoded by the feature +passive) signals the parser that the predicate must have a post-verbal NP at the Deep Structure level, and so the parser moves the NP previously unattached at the Deep Structure level into post-verbal position in the Deep Structure tree. The reader may verify that this parse of a non-reversible passive takes exactly the same number of steps as the parse of a corresponding active sentence, precisely the Slobin result.

In the case of reversible passives there are no corresponding selectional cues to tell the parser to retain the surface subject NP in the input stream of the Deep Structure analysis. Therefore this NP will be attached blindly to the Deep Structure tree just as in the active case. Then, the recognition of the passive morphology when the Aux and V nodes are assembled signals that this NP must now be retrieved and then inserted in proper post-verbal position in the Deep Structure tree, as before. It is this extra retrieval step that constitutes the extra complexity of the reversible passive analysis.<sup>23</sup>

<sup>23</sup>One could easily modify the procedure just outlined to accommodate the Forster and Olbrei (1973) results, should they turn out to be a more valid account of active/passive complexity differences than the Slobin experiments. Recall that Forster and Olbrei observed that *all* passives take significantly longer to parse than their active sentence counterparts. Further, Slobin's reversibility effects were not found.

One could handle a lack of reversibility effects by assuming that subcategorization information is available to the syntactic processor, but not selectional information. (This idea was first proposed in Chomsky [1965].) Then, in the ST model, both reversible and non-reversible passives would retain NPs in surface subject position of the Deep Structure representation, and later movement to post-verbal position could cause extra complexity. Similarly, in an EST model, neither reversible nor non-reversible passives would be marked with a binding index, and so finding the right antecedent would take roughly the same amount of extra time in either case.

The key property of the so-called passive rule that permits us to achieve this speedup by concurrent processing is that passive expresses a *local dependency*. That is, once the passive morphology has been detected, the parser need 'search' only a bounded distance (in terms of phrases) back from the verb to locate the displaced Object. In other words, the passive rule can be expressed in terms of the local, five-cell 'vocabulary' of the parser's transition function, and items in the current cyclic clause that is part of the parser's active memory. Since access to any of the five cells within the parser's purview is presumed to require constant time, given a strict *sequential* execution the total time for the verb attachment and object relocation associated with a passive would be at most some multiple of the primitive execution time of parsing operations, say, three operations *versus* one for the active form. (one to attach the verb; one to locate the NP under the S; one to attach the NP under S as the Object). This multiple cost is 'recoded' into a unit cost by assuming the simultaneous attachment of the verb and the (pseudo) movement of the true Object NP.<sup>24</sup>

More generally, such a result demonstrates that *external, real-world time* (that measured by the experimenter) need not bear any simple relationship to *algorithmic time* (the number of steps used by some procedure under some simple model of computation, like a Turing Machine). In fact, as we discuss in Section 2.4, the time observed *via* reaction time probes may be more closely related to *parallel time*, which corresponds (perhaps counter-intuitively) more closely to the *space* that a *serial* Turing Machine uses.

We may contrast a bounded rule like passive with a rule of grammar that operates over seemingly unbounded domains such as *wh* movement. If the 'speedup' analysis is accurate, then our inability to apply local recoding to a rule like *wh* movement should imply that the processing of *wh* movement sentences should take longer and longer amounts of time as the distance between the displaced *wh* clause and its underlying position grows. In fact, this complexity distinction between local and unbounded dependencies is just that suggested by the ELT theory of Bresnan (1978). ELT proposes to eliminate as transformations precisely those rules that are expressible as local dependencies (passive, there-insertion, so-called raising, and the like), retaining only 'unbounded' rules like *wh*-movement. Intriguingly, just those rules alleged to be 'not realistically captured' by TG are amenable to local

<sup>24</sup> Another way to look at this effect is that it is technically much like the linear speedup theorem of automata theory: by recording two unit operations as one, we can 'speedup' any computation taking time  $cx$  to one taking  $(1 + \epsilon)x$ , simply by expanding the instructional repertoire of the underlying machine to include all finite combinations of actions composed of previously primitive elements. (E.g., (i) *Attach* and (ii) *locate* becomes a single *Attach & locate* operation.)

analysis, hence concurrent speedup in certain parsing models. Thus, the 'non-reality' of these rules can be attributed to particular assumptions about the organization of processing, rather than any failure *in principle* of TG.

### 2.3 A parsing model for the Extended Standard Theory

Like the ST, the Extended Standard Theory (EST) is a model incorporating both a deep and surface structure and a transformational mapping between these levels. However, in contrast to the ST, EST holds that *both* phonetic and semantic interpretations can be read off a *single* representation, namely, Annotated Surface Structure (S-Structure). In ST, the only way to recover the thematic relations of a given Noun Phrase (e.g., *John* in *John was kissed*) is to recover the deep structure. This is because the information that *John* functions as the Object of the predicate *be kissed* is only accessible at the level of deep structure. In an EST framework, when a category is moved it leaves behind a structural residue, a *trace*, in the position from which it originated.

The work a parser must do is, of course, just the reverse of this: it must 'undo' the effects of movement by determining where traces are in the input stream (sometimes a non-trivial task, since traces have no phonetic content), linking displaced constituents to the traces in the appropriate way. Such a parser starts with the representation of a sentence in phonetic form (PF) and derives an Annotated Surface Structure (S-structure) representation. The S-structure then provides an initial format for semantic interpretation, whatever that may be.

Marcus' parser builds a close variant of the EST Annotated Surface Structure and so provides a ready-made format in which to illustrate the potential for concurrent processing. Specifically, wherever a *trace* is required in the parse tree, the Marcus parser creates and attaches a (Noun Phrase) node labelled *trace*, co-indexing the trace to the constituent to which it corresponds. For instance, the output representation for the passive sentence, *John was kissed by Mary* would look like,

[[John]<sub>NP<sub>i</sub></sub>[[was kissed][trace]<sub>NP<sub>i</sub></sub>]].

If a based-generated active form has no movement rules applied (hence no traces), its analysis *via* the Marcus EST parser will proceed just as in the parsing version of the ST model; the annotated surface structure tree will look just like the surface structure tree built by the ST parser. Likewise, the EST parse of the passive sentence *John was kissed by Mary* parallels the construction of the ST surface structure tree up to the point after the passive verb morphology *was sung* is detected in the input stream and labelled as

*passive*. Then, instead of next attaching the verb form and locating and moving the Noun Phrase *John* as in the ST analysis, we shall assume that the parser attaches the verb and simultaneously places a dummy element, a co-indexed *trace*, into the input stream:

Representation in the input stream: |NP-trace<sub>1</sub>| by |Mary|

Crucially, the NP now in the input stream (the trace) will be syntactically analyzed just as if it were a true lexical NP—i.e., as if it were any ordinary Noun Phrase like 'John' or 'Mary'. As a result, it will be attached in the next step as the syntactic object of the verb. The remainder of the parse proceeds as in the ST model: the *by* phrase is parsed and attached to the annotated surface structure tree, as required.

Reversibility effects are relegated to the binding component in the EST model. Just as before, in non-reversible passives one can claim that the mismatch of selectional features triggers an annotation of the surface subject NP with an index indicating to the parser that this NP must be bound to an element somewhere else in the input string. Let us also assume that this binding feature 'percolates' to the S node. Then, when a trace is ultimately to be dropped into the buffer, its binding can *immediately* be established. In contrast, in a reversible passive case the surface subject NP has no selectional feature to trigger the binding annotation. After the trace is dropped back into the input buffer, the parser will have to search back through a portion of the already-constructed tree structure to find an NP to serve as the proper antecedent of the trace. This search may be assumed to be what adds to the observed reaction time complexity associated with reversible passives.

## 2.4 Two views of cognitive capacity

Let us summarize what has been discussed so far. The DTC consists of a conjunct of three hypotheses: (1) a certain type of grammar (e.g., a transformational grammar); (2) a transparent relation between grammatical and processing operations (e.g., grammatical rule types and operations mirrored by parsing operations) and (3) a certain computational organization of parsing. As a result, there are at least three ways in which one could modify the grammar-parser relation in order to accommodate the DTC results. First, one might retain transformational grammar and relax the strict type-type correspondance between grammar and processing units. This possibility will be further discussed in Section Four. Second, one could leave the theory of grammar more-or-less untouched and make the parser give way, changing initial assumptions about available computational power; this was the

approach taken above. Surprisingly straightforward modifications suffice to accomodate the parser to the reaction time data. One simply assumes that the parser is able to perform a (small) finite number of grammatical operations simultaneously, rather than just one at a time, and a rule like passive 'costs' roughly the same as any other (bounded) rule.<sup>25</sup> (In the case of the Extended Standard Theory, this parallelism amounts to being able to carry out two grammar rule 'actions' at the same time. It does *not* hinge upon there being two separate structural descriptions i.e., concurrently constructed deep and surface structures.)

Note, however, that the concurrency model does not necessarily imply that a multi-component (simultaneous) operation like 'passive' takes *no* computational effort.<sup>26</sup> The associated operations are more complex, but the difference in complexity is not measured in terms of time, but rather in terms of the extra 'hardware' that is engaged to effect the computation. By expanding the computational power of the processor, increasing the amount of work we can get done per unit of externally measured time, we can make a once time intensive computation less so. Crucially, in the case of passive we need not expand computational power in an unlimited way: we have assumed only that one can now perform a (small) *finite* number of operations per unit time instead of just one. This is a quite modest use of the computational power of parallelism; as we shall observe below, the general use of parallel machinery admits much broader variation in the apparent time complexity of computations than this.<sup>27</sup>

<sup>25</sup>This definition of concurrent computation should not be confused with others that are current in the psycholinguistic literature. For example, Cooper and Cooper (1980) raise the issue of 'parallel computation'. But for them 'in parallel' means that parsing is not strictly top-down, i.e., that one can build a clause and the dependent of a clause simultaneously. The Marcus parser is in fact not strictly top-down in this sense. Whether the Marcus parser actually parses top-down is a separate issue from whether it can compute two levels of representation concurrently (the sense of 'parallel' that is exploited in the model above). These different interpretations lead to some difficulty in interpreting experiments purporting to show that people can or cannot compute 'in parallel'; we discuss this matter later on in this section.

<sup>26</sup>We have chosen the case of passive for concreteness but the same logic applies to the realization of many other rules investigated under the rubric of 'DTC experiments'.

<sup>27</sup>It has been suggested to us by Bresnan (personal communication) that certain 'feeding relationships' between the NP movement rules will cause problems for the speed-up procedure suggested above. The problems occur, for example, when the passive interacts with dative or when there are interactions between *there*-insertion or any of the other NP movement rules within a single cycle. A relevant case is the following: *John was given a book*. The problem is, in our terms, that once we re-insert the trace of 'John' in its immediate post-verbal position, we still must represent the fact that it is in a dative structure. We agree that this *might* cause problems, if we were to interpret the dative as an active computation, because in this case we would have to wait until we undid the passive before we could undo the dative, thus adding time complexity. This assumes that the parser must mimic the operations of a transformational grammar in a one-for-one fashion. This is not the only possibility,  
(continued overleaf)

Third, and in contrast to the parser-modification approach, one could proceed as in Bresnan (1978): fix a particular view of human computational abilities, and then modify the grammar so as to comply with the psycholinguistic evidence. Instead of assuming deeper computational power, this approach removes supposedly costly operations such as passive from 'active' processing and replaces them with the retrieval of lexical forms. These 'forms' can be thought of as reproducing the *effect* of rules like passive: instead of having a single lexical entry for each passivizing verb plus a single passive rule to generate (or recognize, if we are parsing) corresponding passive lexical forms, Bresnan (1978) substitutes two separate lexical forms for each verb.<sup>28</sup> Thus the effect of the passive rule is 'precomputed' by expanding the rule over all verbs in the lexicon before any sentence is processed. Implicit in this view is the assumption that it is easier to look up a pre-computed result than to compute it using some rule, in short, that memory storage is large and retrieval is fast (nearly costless):

---

however. For example, one can imagine a system in which rules of 'passive' and 'dative' are acquired separately, and yet can be combined by a mechanical procedure to yield possible interactions: these interactions are then 'sped-up' to unit time complexity. Note that now the parser will make *finer* distinctions than the grammar, namely, it will have a rule of 'passive-dative'. There would be no such rule in the basic system of knowledge that is *acquired*, however. In this particular case, how language is put to use would be a refinement of 'knowledge of language'.

However, the principles of TG, or at least recent versions of TG-based theory, lead one to the conclusion that the dative is in fact to be treated as a lexical rule. The first empirical argument in a TG framework to this effect is to be found in Oehrle (1975). Moreover, unlike passive, a transformational dative rule would violate most otherwise well-motivated principles that govern the class of transformational rules as a whole. For example, another set of arguments that would rule out a transformational treatment for the dative in English may be found in Dresher and Hornstein (1979). Dresher and Hornstein motivate the 'trace erasure principle', that states, 'only designated NP elements [like "it" "there", *rcb/asw*] can erase traces'. A dative rule would, of course, violate this principle. (The trace erasure principle is subsumed in the more recent Government-Binding Theory [Chomsky 1981] by the so-called theta criterion and the Projection Principle and so the transformational treatment is ruled out by this theory as well). The point is that since passive and dative must be distinguished by the principles of a TG-based grammar, we are free to treat the dative as a lexical template while treating the passive as an active computation. We think that the other relevant rule, *there*-insertion, will also not cause problems, because here the parser has an overt cue (the designated element *there*) to tell it to expect to find a displaced NP category in the upcoming input stream. See Weinberg, 1979 (Reference note 6).

<sup>28</sup> As far as we can tell, this also holds true of the theory espoused in Kaplan and Bresnan (1981): there is one lexical entry for the active form of a verb and one lexical entry for the passive form. It is quite easy to show that in this case the recognition time complexity for some ELT languages must still be so hard that there is (at present) no *known* (serial Turing machine) algorithm for recognition that runs in time that is less than exponential in the length of input sentence lengths; further, it is highly unlikely that there is a polynomial-time algorithm for recognizing such languages (see Berwick, 1981a, 1981b [Reference note 1]). Interestingly enough, a modestly-restricted ST theory can also be shown to generate languages that are recognizable in exponential time (Kounds, 1975). It should also be pointed out that the Kaplan and Bresnan (1981) restrictions already go far beyond those proposed by Rounds. It remains to be seen what the recognition time complexity is for a comparably restricted transformational theory.

Finally, I assume that it is easier for us to look something up than it is to compute it. It does in fact appear that our lexical capacity—the long-term capability to remember lexical information—is very large (Bresnan, 1978 page 14).

The underlying assumption here—that memory capacity far exceeds computational capacity—is what leads directly to a theory of parsing that attempts to maximize the engagement of memory resources relative to those of computational operations.

We thus have two divergent views on human cognitive capacity as it is employed in sentence processing. On the one hand, the parser-modification approach suggests that the real-time computational power of the language faculty is possibly quite deep, and that significant resources are available to effect rapid calculations for parsing. On the other hand, the Bresnan (1978) proposal implies that what can be rapidly computed is quite limited, and that therefore one must rely on previously stored 'remembrances of words past'. These two views seem to be empirically indistinguishable, at least for the restricted domain of psycholinguistic results for which any comparisons are available. We might, however, uncover other empirical reasons for choosing one approach over the other. For instance, the amount of parallel computation required might be beyond any reasonable upper bound on human computational capacity.

However, this possibility is probably unfounded, for two reasons. First of all, if one pursues the notion of parallel computation, one can show that the resources required for the modest parallelism described in the previous section are physically feasible. Second, if one looks at other cognitive domains where rapid processing is at a premium and where we have some hard knowledge about the associated neural 'implementation'—notably, early visual processing (though there are other examples)—we find that the neural hardware involved actually implements parallel computational power far beyond that required for syntactic analysis.<sup>29</sup> We should of course be

---

<sup>29</sup> For studies of early visual processing, see the work of Grimson and Marr (1979), Marr and Hildreth (1980), Marr and Poggio (1977, 1978), Richter and Ullman (1980), Ullman (1979) and many others. It now seems reasonable to suppose that the (primate) nervous system's computational solution to such problems as finding the edges of objects, detecting motion, and matching points from left and right retinal images so as to obtain a fused stereo image (and hence depth information) all involve a rich, highly parallel network of nerve cells that is inter-connected in a quite specific fashion to compute exactly those functions demanded of it according to the theoretical account.

For examples of 'memory driven' motor control computation, see Horn and Raibert (1978). For recent proposals that show how changes in representational format may eliminate the need for memory driven motor control, see Armstrong (1979), Luh *et al.* (1979) and Silver (1981).

Interestingly then, the history of scientific investigation in two domains—early visual processing and motor control—has been roughly the same. In each case, the very first computational accounts  
(continued overleaf)

extremely cautious about generalizing these findings across cognitive domains; the computational problems the visual system solves presumably do not include syntactic analysis of the sort required by the language faculty. But we can conclude from an examination of the powerful hardware in the visual system that such computational power is available at least in principle to the language faculty.<sup>30</sup>

Let us attempt then to make precise some of the notions of 'depth of processing' and 'parallelism' that we considered informally in the previous section. First of all, neither notion makes much sense without some *reference model* of computation. Otherwise, we cannot properly compare the resource use of one procedure relative to another. Given the central role of a model of computation in the evaluation of procedures with respect to their resource use, it is not surprising that much effort has gone into showing that resource evaluation is relatively invariant under large shifts in reference models. For example, one can show that the number of steps (= 'time') that a procedure takes on a Turing Machine is within a small polynomial factor of the number of steps the procedure would take on a much more 'realistic' model of a modern computer, a Random Access Machine (RAM).<sup>31</sup> This means that the Turing Machine model, idealized as it may seem, is just as 'realistic' as the Random Access Machine model for the purposes of evaluating time complexity—if the evaluation ranking stays the same under small polynomial variation.

It would seem then that the choice of a reference computational model for the purposes of cognitive investigation could depend on at least two factors: (1) known *invariance* results: Does it matter what model we pick, or does the evaluation ranking stay fixed across models? (2) *empirical considerations*: Does the range of models considered cover the possibilities

---

of how a particular human competence should be 'realized' involved a reliance on looking up 'pre-computed' answers, remembering past visual, auditory, or motor control 'templates'. In each case, 'look up the answer' theories are now in competition with theories that assume a richer computational power, often schemes that are more closely tailored to the particular domain under investigation. We would not be surprised if the investigation of syntactic processing turns out to recapitulate this history. However, it remains an open question as to whether memory-intensive or computationally-intensive methods are used in linguistic processing; one's initial guess is that both methods are used.

<sup>30</sup>Note that we are not saying that the trade-off between large-scale memory storage *versus* rapid, deep computation has been settled against memory retrieval and in favor of 'deeper' computation; rather, we intend to point out that there is a trade-off to be made, and that memory retrieval is not *always* the answer to the need for 'real time' computation of a difficult problem. We have simply shown that *neither* method can be dismissed *a priori*; both provide a reasonable architecture to underpin models of language use.

In passing, we should point out that it is somewhat confusing to distinguish memory retrieval from 'active' computations, as if memory retrieval did not itself involve computation of some sort.

<sup>31</sup>This is a standard result; see, e.g., Machtey and Young (1978).



for the human cognitive system under study? It is by no means obvious that the 'right' underlying model of the computation underlying cognitive systems should be the 'usual' serial Turing machine or even a Random Access Machine. More 'exotic' models, such as the parallel scheme described earlier, are also candidates.

It is, of course, widely known (or suspected) by psychologists that the introduction of 'parallel computation' might radically alter one's view of what is or is not easy to compute; it is an informal aphorism that parallelism naturally allows one to compute faster. For example, consider a recent series of articles debating the possibility of finding evidence to distinguish between propositional and imagistic theories of mental representation (Anderson, 1978, 1979; Hayes-Roth, 1979; Pylyshyn, 1979). In his reply, Anderson rebuts the point that the propositional mimicry of an image might take exponential time (and hence perhaps be distinguishable from an imagist theory on the grounds of a detectable increase in externally measured processing time) by invoking parallel machinery: he cites a result catalogued by Meyer and Shamos (1977)<sup>32</sup> showing that a simple model of parallel computation, a *Boolean circuit network* of and and or gates allows one to enormously speed-up computations that take exponential amounts of time by expanding the number of primitive operations that can take place simultaneously. This result is a familiar story to computer scientists: by expanding the amount of hardware or space allowed, one can often reduce the amount of time it takes to compute a given function. But we do not have in mind this standard (and straightforward) demonstration of the interchangeability of time and space resources. What is apparently less well known are the following two results. First, the *standard* theorems adduced to demonstrate the power of parallel speedup hold only if one posits computational circuitry that is *not necessarily* physically realizable.<sup>33</sup> Second, notwithstanding the difficulty of translating mathematical results to the real world, a similar kind of radical exchange of time for space *does* still hold for models that assume physically constructible parallel devices (in fact, for all such reasonable parallel models that have so far been proposed).

In a network model of and and or gates, parallelism is captured by being able to perform any finite number of gate operations (ands and ors) at any single step. This formalizes the sense of 'doing more than one thing at a time' that we talked about informally in the previous section. The total time (number of steps) that the computation will take is clearly equal to the

<sup>32</sup>Based on work by Borodin (1977), Pratt and Stockmeyer (1976) and others.

<sup>33</sup>These are chiefly assumptions that one is able to expand the required hardware and its associated wiring arbitrarily, or, perhaps, manipulate in parallel vectors of numbers of arbitrary length.

length of the longest path in the circuit from input to output; this is the *Depth* of the circuit. Depth is one natural complexity measure of the computational work done by a circuit, one that corresponds to a measure of *parallel time*. We can also measure the complexity of a circuit by the amount of 'hardware'—number of gates—required to construct it. By convention this is called the *Size* of the circuit.

These two measures—circuit depth (parallel time) and circuit size (amount of hardware)—relate in an obvious way to the methods we used to speed up parsing time at the expense of increased parallelism. For instance, recall the demonstration that the rule of passive could be incorporated into a fast parser if we could expand the amount of work allowed at any single 'step'. This is just the sort of expansion that is captured by a circuit model. The circuit also permits the amount of hardware required to compute the answer for any particular input to vary—we are allowed to use more gates to parse a sentence ten words long than the gates we use to parse a sentence of length, say, five. A trade-off between parallel time and hardware thus arises quite naturally in the context of the circuit model: by changing the 'wiring diagram' of our machine we can effect a substantial speed-up of certain computations. If the neural circuitry is faithfully mirrored by such a model, then it is at least possible that the same sort of parallel speed-ups are exploited there as well.

This possibility poses a specific problem for those who have already fixed upon a serial computational organization as the 'right' underlying model with which to judge an algorithm's complexity. For suppose that the assumption of seriality is incorrect, and that the function is realized using parallel circuitry. Then the time it will take to compute an output will be mirrored by circuit *depth*. If this is so, then a reaction time probe—a measure of external clock time—will measure circuit depth and hence parallel time. This makes sense: if an operation is underlyingly parallel, then its execution time as measured externally (let us call this *phenomenal* time) should be identified with *parallel* time (let us call this *algorithmic* time).<sup>34</sup>

---

<sup>34</sup> As discussed immediately below, one should not confuse this kind of synchronous parallelism with the perhaps more familiar case of asynchronous parallelism. In asynchronous parallelism, two or more quite different (typically intermodal) tasks are carried out simultaneously. The speed-up results refer only to the former sort of concurrency, where a *single* function has been designed from scratch to be computable in parallel. This is just the sort of parallelism involved in the passive analysis discussed in the previous section.

Further, one should realize that there is no paradox between the inherent seriality of the speech stream and the possibility of parallel computation over those tokens. The Marcus parser is designed precisely to resolve this paradox: tokens are read groups at a time into an input buffer, and then rules may operate (in parallel, if need be) over the entire set of tokens. There is then a limit to the *amount* (continued on facing page)

The key question can now be properly formulated. What happens if we use the wrong algorithmic model for a computation? Is there any cause for alarm? That is, is there any distinction between serial and parallel time that could cause problems if we confused one with the other?

We would claim that it *is* potentially misleading to use a serial reference machine where a parallel one would be correct. The reason is that parallel time (circuit depth) does *not* map over into serial time (as clocked by a Turing Machine) in the natural way one might expect. A circuit of Size  $T \log T$  can *simulate* a Turing Machine that uses *serial time*  $T$ . ( $T$  is some function of  $n$ , e.g.,  $n^3$ , where  $n$  is the length of the input, e.g., the number of words in the sentence to be parsed, if we are parsing.)<sup>35</sup>

Thus, the 'time cost' of an algorithm with respect to a standard serial computational model need not directly reflect the amount of externally measured *time* it would take a person to carry out the procedure. Rather, the cost might be more closely allied (perhaps counter-intuitively) to the amount of *hardware* (circuit size) engaged. The relation between externally clocked time and algorithmic time is lost because external time maps in a more complicated way to the Size of a circuit. A circuit can often compute the same result as the Turing Machine in less external time by becoming 'wider', thus keeping its Size to within the required  $T \log T$  bound but compressing the needed Depth. The exact compression possible would depend upon the number of gates allowed at any one level of the circuit, the number of wires that could feed into and out of gates (their 'fan-in' and fan-out'), and, of course, exactly what problem was being 'solved' by the circuit. The externally observed time behavior of such circuits could be diverse, ranging from little apparent difference with the serial algorithm to an apparent exponential increase in speed. In short, ticks on the external, experimental clock would no longer necessarily correspond to ticks of the internal, algorithmic one—because we would have unluckily picked the wrong model for timekeeping. This is the abstract counterpart of the situation we discussed in Section One: experimentalists have generally equated externally measured, *phenomenal* time with *serial algorithmic* time,

---

of parallelism that is allowed: the buffer is of finite 'bandwidth'. However, as we demonstrated in the previous section, this limited bandwidth is sufficient to handle the parallelism required, at least for the passive case.

<sup>35</sup>See Borodin (1977), Cook (1980), Ruzzo (1980). Importantly, one demands that these circuits have limited 'fan-in'; that is, at most two wires can enter any gate. In addition, the circuits must be *uniform*, that is, given the circuit required to solve a particular problem of input size  $n$ , it is computationally easy to construct the circuit required to solve the next biggest problem (of size  $n + 1$ ). Finally, by simultaneously limiting the amount of hardware expansion to some polynomial function of the input size, one may ensure that such circuits are 'physically constructible'. In this restricted case, the size-time tradeoff still holds.

assuming that external time complexity for sentence processing must be proportional to the number of grammatical rules involved in the derivation of a sentence. The lack of fit between grammatical model time steps and external time steps has then been taken to imply a weakness in the theory of grammar, and a demand for more suitable theories. However, this conclusion does not necessarily follow; the simplest remedy could be to move to an appropriate parallel clock.

It is important to bear in mind that these speed-up results hold under models of *synchronous* parallel computation, that is, models where the machine has been designed from the start to operate concurrently to compute some function. The parser described in the preceding section is one such machine (though its parallelism is rather modest); it is meant to compute just one basic output representation, with some of its operations taking place simultaneously. These machines should not be confused with models of *asynchronous* parallelism, where several machines, perhaps computing the same function, perhaps not, all compete (relatively) independently for a common set of resources (time or space). A paradigmatic type of asynchronous parallelism is a time sharing system; here, many different programs (run by different users) all vie for central resource use. As users of time sharing systems are well aware, there may be no speed-up at all in some asynchronously designed systems; indeed, it is often the case that as more programs compete for limited resources, the time it takes for each individual program to execute rises dramatically.<sup>36</sup> This distinction is crucial because many of the studies demonstrating that people have rather limited abilities to 'process in parallel' have assumed, probably quite rightly, an asynchronous model of parallelism. This assumption is a natural one since the tasks that have been tested have been intermodal in nature, and thus naturally fall under the 'operating system' rubric of competition for common resources. (See Posner and Mitchell [1967] and Townsend [1974]). The results also agree with the folk experience of time sharing users: parallelism can often degrade performance.

In contrast, synchronous parallelism is most naturally interpreted to operate only intra-modally, *within* some single component like a syntactic parser. The resource competition paradigm is not necessarily appropriate; so results like Posner's indicating the apparent lack of parallel speed-up in people need not apply. Indeed, distinguishing between synchronous parallel and serial computation would seem to be a difficult experimental task, as

---

<sup>36</sup>Even if resources are unlimited, it is possible to prove quite specific theorems about bounds on the speed-up possible when a single problem is solved by asynchronous parallelism. See Kung (1976) for a survey of recent results.

indicated by Townsend (1974). For example, as Anderson (1979) observes, the usual Sternberg additive componential paradigm “analyzes information processing into a sequence of stages, specifies which factors affect the time for each stage, specifies the time parameters for each stage, but does not analyze in computational detail why each stage takes as long as it does or why it is affected by factors in the way that it is” (1979 pp. 404–405). This means that the ‘inside’ of any stage is opaque to further computational decomposition; one is free to choose a serial or parallel mechanism to ‘realize’ the computational guts of any stage, subject only to constraints of externally observable time behavior. But as we have just seen, this means in turn that the time we observe *via* the Sternberg analysis for individual stages might actually be *parallel* circuit time (not Turing machine serial time), and parallel circuit time maps over into Turing machine *space*. If this is so, then models based on serial time would have to be judged in terms of their *space* efficiency, not their time efficiency.

Let us stretch more carefully just how, even under the Sternberg assumptions of linear stage decomposition, parallel computation might make the interpretation of competing processing models more difficult. Suppose that the response time for some isolated stage has been determined to vary as the square of the input problem size,  $n^2$ . Thus phenomenally observed time is quadratic. Suppose further that there are two processing models that have been proposed to account for the computation of this stage, and that the complexity of both models has been evaluated with respect to a serial reference machine. Model A uses quadratic time and linear space; model B uses cubic time and quadratic space. With respect to an assumption of seriality, Model A comports with the psycholinguistic evidence; Model B is too slow. But if a parallel circuit reference base is adopted, then B can run in parallel quadratic time  $n^2$  (its old space requirement); model A, in parallel linear time. The result is that Model B is now closer to the psychological evidence; Model A is now perhaps too fast.

The general lesson to be drawn from the parallel circuit model then is that we must be careful when we suggest that one procedure is ‘better’ than another in a *cognitive* domain. Claims about cognitive capacity and the trade-off between time and space resources depend upon both a precise specification of the algorithms to be compared and an underlying model of computation to be used as a reference base for comparisons. Without confidence in the fidelity of the reference model, any claims of algorithmic superiority may be empty. In particular, if the comparison is made on the basis of any underlying model that is false to the facts—if we claim that one procedure is better for people than another procedure because the first is faster, but this is true only with respect to machinery people do not

have—then we are misleading ourselves. We cannot simply *stipulate* an underlying model of computation on which to base our predictions of the amount of externally measured time a procedure takes without substantial support for that model; otherwise we may be favoring certain kinds of procedures capriciously at the expense of others. Even though it may seem intuitively plausible that it is easier to ‘look things up’ than compute them, it is not a logically necessary solution; in fact, the evidence from the domain of early visual processing points in quite the opposite direction. This simply makes it plain that cognitive science, is, in the end, an empirical, biological science: while we may speculate about what is or is not the ‘right’ computational organization of the brain (based on whatever psychological, engineering, or computational biases we may have), ultimately there is a fact of the matter that can render such speculation moot. If the brain uses parallel computational power to process sentences rather than memory lookup, then that is what it uses; no stipulation can change this fact. If our aim is to discover what computational organization the brain does have, then stipulation again seems unwise. Given our current lack of understanding, it would seem best to keep all (so far indistinguishable) computational organizations available, lest we rule out by fiat the ‘right’ theory of processing.

There is one further point to discuss before concluding our comparison of extended lexical grammar and the Extended Standard Theory. In the preceding discussion we have assumed (along with Bresnan) that the transformations postulated by EST should be thought of as ‘active’, time-consuming computations. We then provided a machine architecture that would make an EST-based sentence processor compatible with certain psycholinguistic results. However, it is not obvious just why it is necessary to make this assumption, unless one is also insisting upon some strong version of transparency, i.e., that the move-NP rule and its parsing correlate must be stated in exactly the same form. If transparency is relaxed, it is possible to embed an ST-or EST-based parser in a serial computational model. One could do this by ‘precomputing’ the effects of the transformational component and storing the results in the lexicon. Note that this in no way disconfirms TG either as a grammar or as a central component of a parsing model; it merely says that the way in which a grammar may be embedded as part of a model of language use is less than straightforward.

In order to understand why this is so we must be sure to distinguish the claims that a grammar makes about the system of knowledge incorporated in the language faculty from the implications of those claims for a theory of parsing. Let us make this point clear with a concrete example. Recall that psycholinguistic results show that passives and actives take less time to parse than *wh* movement constructions. In an ELT, simple active sentences

are derived from a context-free base component, passives are derived *via* lexical rules, and *wh* questions are derived by (structural) movement rules (or their interpretive counterparts). Note that lexical rules and movement rules are quite distinct: lexical rules are governed by 'functional' criteria, while movement (or its interpretive analogue) makes crucial reference to structural principles (phrase structure nodes). This distinction in the *grammar*—plus Transparency—is what warrants drawing a corresponding distinction between these two sorts of rules in the *parser*.

In contrast, a TG treats both passive (or NP movement in current theories) and *wh*-movement as parts of the same component of the grammar (namely, as sub-cases of a more general 'move alpha' rule in the most recent approaches). Consequently, certain critics of TG assume that one cannot draw any formal distinctions between NP and *wh*-movement. Therefore, given a strong version of the Type Transparency Hypothesis, we should not be able to distinguish between these two sorts of rules in the parser; we are not licensed to have one parsing procedure for NP movement and another, quite distinct computational routine for *wh*-movement. The conclusion that follows is that we could not embed TG in a parsing model of the sort Bresnan envisages.

The problem with the implicit assumption is that it is false. Transformational grammar has long recognized that there are important formal differences between NP and *wh*-movement. For example, Emonds (1970) argues that NP movement rules are *structure preserving*,<sup>37</sup> while *wh*-movement rules are not.<sup>38</sup> Moreover, even in the most recent theories where NP and *wh*-movements are literally taken to be special cases of a more general 'move alpha' rule, the theory postulates different relations between a moved element and its trace. For example the trace of NP movement cannot be case marked, while the trace of a *wh*-movement must be in a case marked position. Therefore, there *are* criteria to distinguish NP movement from *wh*-movement in the grammar. We are, then, licensed to precompute the effects of NP movement and store them in the lexicon, while continuing to 'realize' *wh*-movement as an active computation—even in a TG based parser. Given the assumptions of TG however, one should note that the precomputed lexical templates associated with NP movement would necessarily be governed by purely *structural* principles, unlike the templates of extended lexical grammar.<sup>39</sup>

<sup>37</sup>That is, NPs can only be moved into phrase marker positions where NPs can be generated by the phrase structure rules of the grammar.

<sup>38</sup>See Emonds (1976) for an alternative analysis; but also see Freidin (1978) and Chomsky (1977) for arguments that Emonds' first hypothesis was correct.

Let us summarize the conclusions of this section. First of all, there seems to be no reason to adopt the particular machine architecture envisioned in Bresnan (1978). The psycholinguistic evidence so far adduced does not force this choice, and indeed there is some reason to believe that an alternative computational organization would be more faithful to what is possible, given human computational machinery. Second, there is no inherent connection between the choice of a 'functional' grammar (like that of Bresnan [1978]) and the decision to favor memory retrieval to account for the processing of active and passive sentences. Both TG and ELT provide the necessary principles to distinguish among rule types so as to permit some rules to be handled by memory retrieval and others by active computation, all within a machine architecture like that assumed in the DTC or in Bresnan (1978).

### 3. Grammars and parsers: the autonomy of syntax thesis

In our discussion of DTC and its subsequent revision in Bresnan (1978) we observed that the Type Transparency Hypothesis was invoked in a model equating psycholinguistically observable time complexity to a simple model of algorithmic time complexity, and this, in turn, to a measure of derivational complexity in the grammar. In this section we examine work by Tyler (1980) and Tyler and Marslen-Wilson (1977) that also uses the Transparency Hypothesis to relate psycholinguistic observables to the theory of grammar.

---

<sup>39</sup> Actually, the story is more complicated than this. Bresnan (personal communication) points out that in a theory like TG that allows successive cyclic movement into complementizers one might think that the *wh*-element in COMP would also be in a strictly local domain, and one could thus employ the speed-up technique in this case as well. Under these assumptions we would not capture the purported difference between *wh*-movement and passives with respect to psycholinguistic timing results. To enforce this distinction, should it be relevant again, we stress that Slobin's results are extremely controversial and that under the assumptions of Forster and Olbrei (1973) we would expect *wh*-movement and passive to take the same amount of time—we must appeal to another principle as well. This is a principle motivated primarily by parsing considerations that stipulates that only completed constituents may be attached to nodes stored in the Marcus parser's active node stack. Thus in the relevant cases, before one could bind a *wh*-element to its trace one would have to attach all the material to the S node dominating the trace, then attach this node to the S-bar. Only after this attachment has been made could one bind the trace to the *wh*-element in the S-bar. This is because, strictly speaking, the grammar rules are only sensitive to material in the immediate dominating cyclic node (NP or S). See Marcus (1980) for details. Because the parser must wait for this attachment before binding takes place, we might expect timing distinctions between *wh*-movement and passive cases. It should be noted that the decision to attach S and S-bar separately and to govern the attachment of S to S-bar by a principle that says, "first complete constituents, then attach them" is not an arbitrary decision. There are cases where it is crucial that this procedure be followed: see Marcus (1980) for details. Moreover, this assumption of uniformity figures crucially in the over-all learnability of the system (see Berwick, [1982, Reference note 2] for details).



As mentioned in the introduction, Tyler and Marslen-Wilson try to show that the autonomy of syntax thesis is untenable as a principle of a grammar *that one could embed in a model of language use*. Their argument rests on two assumptions. The first is the claim that under Type Transparent assumptions about the relationship between the grammar and the parser, the autonomy thesis implies a serial organization between the components of a TG. This assumption is not original; rather, it seems to be widespread in the psycholinguistic literature. Then they show that this organization must be abandoned in the light of several experimental results. Our argument against this position consists in trying to refute the first assumption. We will show that even under such Type Transparent assumptions, the autonomy thesis says *nothing* about the flow of information between components. Rather, it spells out the types of information that can form the representations of each component. Therefore, for Tyler and Marslen-Wilson's argument to go through, they would have to show either that the representations of each component do not respect the autonomy thesis, or that by setting up the representations in this way, we become unable to build a reasonable parsing algorithm. In fact we shall see that the autonomy of syntax thesis is perfectly consistent with the algorithm that they themselves envision.<sup>40</sup>

Let us begin by outlining Tyler and Marslen-Wilson's position. Tyler (1980) begins with a review of the literature that suggests that realizing a transformational generative grammar (TGG) imposes very special conditions on parsing implementations. Any parser that does not exhibit such an implementation cannot be said to "realize a transformational generative grammar". To be specific,

Any processing model which intends to maintain its links with TGG must adhere to these two principles...autonomous syntax and the delay of sentential semantic analyses until a syntactic deep structure representation has been assigned (Tyler, 1980, page 9).

If we grant the first assumption, it then follows (as Tyler and Marslen-Wilson [1977] claim) that the validity of their processing model (which adheres to neither of these two principles) must

...cast doubt upon the viability of using a TG as a basis for a psycholinguistic processing theory... The more radical interpretation that semantic and syntactic analyses continuously interact as a sentence is heard (an interpretation which they

---

<sup>40</sup> As we noted above, Tyler makes this assumption about the relation between grammars and parsers explicit when she says: "Although the linguistic models did not themselves, strictly speaking, consist of processing components standing in some ordered relationship to each other, the psycholinguistic interpretation of these grammars required them to be treated as if they were making claims about the order of processing events" (Tyler, 1980, page 2).

endorse rcb/asw) is incompatible with even a modified transformationally-based processing model (op. cit., 1977, page 650).

This leads to a view calling for the strict separation of the study of linguistic competence from the study of linguistic performance.

In evaluating these claims, we must ask the following questions: (1) Does a TGG entail the kind of implementation implicit in assumption 1? (2) If not, what sorts of implementation would a TGG be consistent with? We should first note that Tyler's conception of the autonomy of syntax should be understood as autonomy of syntactic *processing*. Tyler borrows a definition provided by Forster (1974) to explicate this notion. Autonomous syntactic processing means "... semantic processing is delayed until intact deep structure units have been isolated, and in general, considerations of meaning are irrelevant to syntactic decisions" (Forster, 1974, page 391).<sup>41</sup> Autonomy of syntactic processing (*qua* Tyler) means that (1) linguistic information is compartmentalized into (at least) two separate components, one syntactic and the other semantic; and (2) semantic interpretation can only begin once the syntactic component has completed its analysis and generated an output representation; semantic processing cannot start until syntactic processing is finished.

The flowchart scheme and componential organization implied by this 'autonomy of processing' thesis is consistent with the way in which transformational grammars are traditionally written. However, this notion of 'autonomy of processing' is unrelated to the autonomy of *syntax* thesis as it is defined in linguistic theory. The autonomy of syntax thesis is defined in Chomsky (1977) as follows:

We can distinguish then, two versions of an autonomy thesis: an absolute thesis, which holds that the theory of linguistic form including the concept 'formal grammar' and all levels apart from semantic representation can be fully defined in terms of formal primitives, and a weaker version which holds that this is true only conditionally with certain parameters, perhaps localized in the dictionary (page 42).

Chomsky (1979) goes on to say that:

... Thus the viewpoint of this work was that, given a linguistic theory the concepts of grammar are constructed on the basis of primitive notions that are not semantic (page 139).

---

<sup>41</sup>The condition that semantic analysis be delayed until full deep structure representation is recovered is an additional condition on autonomous syntactic processing. As recognized by Tyler the autonomy of syntactic processing would not be violated if, e.g., a string was first segmented on-line into syntactic phrases and then semantically interpreted phrase by phrase.

The autonomy thesis, in short, is a claim about the *kinds* of knowledge that a speaker/hearer has. Therefore, this thesis says nothing about the actual analysis of a sentence by a processor. In particular, the syntactic component of a parser is not restricted to operate without making reference to the information provided by some semantic component. The autonomy of syntax thesis claims merely that *however* the computation is effected, the output of sentence processing yields at least two types of information, neither of which is reducible to the other. No claim is made about whether the entire deep structure or parts of it are available on-line for semantic interpretation.

Therefore, the first assumption underlying Tyler and Marslen-Wilson's claim is simply false. As long as a parser encodes as (distinct) types of information the different levels of information provided by the grammar, that parser is respecting the autonomy of syntax, as defined by TGG.<sup>42</sup>

Tyler (1980) and Tyler and Marslen-Wilson (1977) then try to show that the autonomy of syntactic parsing thesis must be abandoned. We will consider only one of these experiments in detail; however the same comments apply more generally to the other experiments.<sup>43</sup> Their experiment consisted in asking subjects to repeat the last word of a deep structure

---

or a similar argument, see Dresher and Hornstein (1976, pp. 331–332), and Valian (1979) (in *per and Walker*, 1979, page 19).

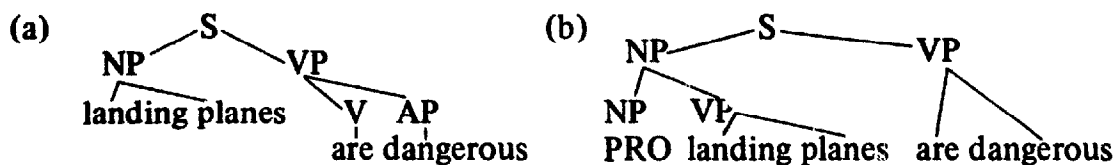
It should be noted that even though the autonomy of syntax thesis *does not imply* the autonomy of parsing thesis, it may still be empirically true that the parser processes autonomously. If we discover this to be the case, we will have found something interesting. We will have found strong principles, not implied by the grammar, that constrain the organization of the parser. There are, of course, other alternatives to autonomous processing; one is to propose that the grammatical components of the parser may interact with each other at random points during a parse. However, as Forster (1979) points out, it would seem methodologically preferable to adopt the former model as a first hypothesis because: "It seems... that if we begin by postulating such a model (the random interactive model rcb/asw), then there is very little hope of discovering interesting structural properties at all, and consequently, we would be reduced to merely noting and cataloguing the kinds of problem-solving strategies that are... employed in various kinds of tasks. This may ultimately be the correct view to adopt but it seems preferable to first thoroughly explore the alternatives to this view" (Forster, 1979, page 36).

<sup>43</sup>Indeed it seems that the case we discuss here is Tyler and Marslen-Wilson's least controversial. We note that in their other cases one of the intended interpretations involves the binding of pronominals, and thus it may be that *binding differences*, rather than access to a semantic component, leads to the different reaction times observed by Tyler and Marslen-Wilson. In fact, Fodor (personal communication) informs us that this point has been noted independently and that research is being carried out to see whether this observation casts doubt on the Tyler and Marslen-Wilson results. We take no stand as to whether these experiments will actually show that Tyler and Marslen-Wilson have conflated dependent variables. Tyler (personal communication) claims that the variables are in fact not conflated. We choose to deal here with Tyler and Marslen-Wilson's least controversial case because we are interested in making a claim that is stronger than showing that they have merely confounded variables in their experimental data. We are interested in showing that *even* if this data is the result of semantic-syntactic interaction, it still is not relevant to their theoretical claims about the relationship of TG to parsing theory.

ambiguous phrase. Subjects heard the context clause and the initial portion of the ambiguous string. Then there was a pause, after which they were asked to repeat the verbal portion of the phrase (capitalized and underlined in the text) that had subsequently been displayed to them. As it turns out, the plural form 'are' is the appropriate continuation only for one reading, (a). Compare:

- (a) If you walk to too near the runway, landing planes ARE  
 (b) If you've been trained as a pilot, landing planes ARE

The two possible representations of the ambiguous phrase are:<sup>44</sup>



The computational complexity of this task was taken to be reflected in the latency between visual presentation of the underlined word and the onset of pronunciation by the subject. By these criteria, (b) was judged more complex than (a).<sup>45</sup>

Tyler and Marslen-Wilson interpret this result in the following way: First they note that only one of the two possible structural representations (the representation where *landing planes* is analyzed as a Noun Phrase) takes a plural verb. In contrast, the second, sentential representation takes a singular verb. The context clause preceding the fragment *landing planes* acts to semantically bias the interpretation of this phrase: in (a) the nominal analysis is favored and in (b) the sentential analysis is preferred. The plural probe ARE is then only an appropriate continuation of the (a) sentence. Tyler and Marslen-Wilson claim that it is the perception of the probe word in the (b) example that causes the longer naming latency.

However, for the biasing effects to have been effective, they also assert that:

The listener in this experiment needs not only to have interpreted the context clause, but also to have combined this with the individual meanings of the words 'landing' and 'planes' into some unified higher-level relational representation.... (Ibid. page 45).

The biasing effect is perceivable only at the 'semantic level' because the well-formedness conditions (subcategorization, selectional restrictions) allow

<sup>44</sup>Here, PRO is taken to mean a non-phonologically realized element that refers to 'pilot' in the context clause.

<sup>45</sup>The latency for (b) was 555 msec.; for (a) it was 519 msec (Tyler, 1980, page 45).

both analyses of *landing planes* to be continuations for each context clause. Therefore the fact that the biasing clauses were effective

... indicates that semantic variables can interact with ongoing syntactic decisions. The preference for a singular or a plural verb form in this task reflects the outcome of a syntactic processing decision. And this decision (the decision of which structural analysis is associated with an on-coming input stream rcb/asw) depends on the on-line interaction of semantic context in the syntactic interpretation of the ambiguous fragment (Ibid. pages 46 and 47).

It is this interaction of semantic interpretation with ongoing syntactic decisions that violates the 'autonomy of processing' hypothesis.<sup>46</sup>

Let us assume for the moment that Tyler and Marslen-Wilson's evidence disconfirms the autonomy of *processing* thesis. By the argument given above, we must still establish whether the interpretation of the experiment violates the autonomy of *syntax* thesis, and it is only this question that is relevant to determining whether the parser is operating in accordance with principles specified by TGG. Recall that the autonomy of syntax thesis specifies the kinds of knowledge a speaker needs. (A speaker possesses at least two separate systems of structural and semantic knowledge based on two different sets of primitives and rule types). TGG also specifies that the semantic component acts to provide an interpretation to the formal output of the syntactic component. Let us then ask what kind of knowledge we need to impute to the subjects of Tyler and Marslen-Wilson's experiment.

First, by Tyler and Marslen-Wilson's own description, subjects had to compute each of the two possible structural descriptions that were generated

---

<sup>46</sup>It should be noted that Forster (1979) provides an account of the Tyler Marslen-Wilson results that respects the autonomy of processing thesis. Briefly, he proposes a parallel parsing model divided into three autonomously functioning components, ordered as lexical, syntactic, and message modules. The output of each component may feed into the next one above it, and may also be relayed to a 'general problem solver' that uses the information provided to perform various tasks set for it (e.g., naming, lexical decision, and so forth). Forster claims that in the interpretation of the (a) and (b) sentences above: "The syntactic processor may be quite unaffected by semantic context... Two analyses of the ambiguous phrase are provided, and one is discarded when the subsequent disambiguating probe word is presented... However, the message processor chooses one of the interpretations of the ambiguous phrase... only to discover that the subsequent output of the syntactic processor is incompatible with this choice" (Forster, 1979, page 54).

Furthermore he claims that the experiment does not necessarily reflect normal parsing procedures: "With normal presentation of the sentence, we might not expect this (latency rcb/asw) effect to occur. Rather, the message processor might delay the choice between the two possibilities to see whether syntactic disambiguation will be possible. However, when the sentence actually terminates before the disambiguating word is presented,... and when the task obviously requires the subject to try to predict ahead, the message processor may commit itself to an interpretation immediately" (Ibid., page 55).

He then goes on to suggest other alternative hypotheses to explain these results. For a full discussion, see Forster (1979). To reiterate however, the principle of the autonomy of syntax may be reflected in the parser irrespective of whether the autonomy of parsing thesis is a valid parsing principle.

for the ambiguous fragment by the syntactic component, and these representations were computed based only upon structural information. A semantic representation capturing the nominal or propositional character of associated structural representations also had to be computed. These semantic representations were checked for compatibility with the preceding context phrase. Thus we see that Tyler and Marslen-Wilson also assume that their subjects had knowledge of two distinct kinds, both of which were involved in parsing the experimental sentences. Therefore the parsing model they envision respects the autonomy of syntax thesis. Interpreted in this way, their experimental results provide no evidence *pro* or *con* for determining whether TGG is realizable as part of a parsing model.

While it is true that transformational grammars are traditionally written in a way that might imply sequentiality of components, we would claim that this fact is of little interest. A strict construal of the Type Transparency Hypothesis would suggest that there are aspects of the formalism of the grammar that may be taken to imply particular parsing organizations. For example, ELT and EST may suggest different predictions about the complexity of memory retrieval processes *versus* active computations, as discussed in Section Two of this paper. But the autonomy of syntactic and semantic processing principles as they construe it is simply *not* implied, even under the assumptions of Type Transparency, by the autonomy of syntax thesis. The serial componential organization actually used in the literal notation of a standard TG is merely a convenient way of writing things down on paper. The *logical* separation of syntax and semantics need not imply that syntactic processes are not able to make use of semantic knowledge during the course of parsing. Under Type Transparency the autonomy thesis would imply that two systems based on different principles are involved in syntactic and semantic comprehension. We could just as easily have captured this autonomous logical decomposition using the diagram that is part of the non-autonomous implementation argued for by Tyler and Marslen-Wilson. When evaluating how direct a connection exists between the algorithms of a parser and the rule systems of a grammar we must be sensitive to the difference between notational variants as opposed to notations that actually imply different principles and parsing effects.

Tyler (1980) proposes what is at first glance a more radical interpretation of the Tyler and Marslen-Wilson results, one that seems to sharply distinguish the parsing system she envisions from those incorporating a TGG. She claims that, by showing that the different components of the grammar may interact at any point in the derivation, we have eliminated the need to postulate separate representations to serve as the outputs for each level:

If the listener could draw simultaneously on both syntactic and semantic knowledge to construct two interdependent representations, then these two representations would hardly be functionally distinguishable and the separation between them would appear to serve no independent purpose (Tyler, 1980, page 55).

She further suggests that a model incorporating a TGG should be replaced with a system organized along lines proposed by Winograd and his fellow AI researchers:

What such AI systems illustrate for us is just that it is indeed computationally possible to construct language processing systems which do not maintain the kind of separation between syntax and semantics that is central to TGG and to the processing models based on the TGG framework (*Ibid.*, page 57).

Let us begin the examination of this proposal by asking the same question that we have been asking throughout this section. Does this proposal respect "the kind of separation between syntax and semantics central to TGG" or does it incorporate a grammar that violates the autonomy of syntax thesis?<sup>47</sup>

The answer to this question very much depends on what we mean when we say that representations are no longer 'separated in parsing'. Tyler seems to vacillate between two positions. One thing that she could mean is that the grammar incorporated in the parser no longer distinguishes purely structural from purely semantic information, and that rather than having two separate systems, each based on a different set of primitives and principles, we in fact have one system that can be characterized by a uniform set of primitives. In such a system we could claim that either general intelligence or general linguistic strategies carried out the work that was formerly handled by distinct TGG rule systems. Another way of putting this is that one could now retrieve the syntactic form of a representation simply by knowing its semantics. There would be no such thing as 'semantic' or 'syntactic' rules; there would simply be *rules*, all guided by the same principles, that produce a single syntactic, semantic, and phonological representation. This picture is surely at odds with the autonomy of syntax thesis. It also cannot be what Tyler and Marslen-Wilson have in mind. As mentioned, in order to explain the data in their experiment they impute to their subjects knowledge based on two separate systems: they attribute the capacity to compute a structural description for an input string and the capacity to interpret that string semantically.

This position also seems to be a very unlikely view from the processing perspective. Forster (1979) has shown that careful experimentation can

---

<sup>47</sup>It should be noted that the Tyler and Marslen-Wilson experiments do not necessarily lead to this conclusion. See Tyler and Marslen-Wilson (1979) for a model of interacting components that, even under Type Transparent assumptions, respects the autonomy of syntax thesis.

provide evidence that the different processing levels can be separated out from each other. We can find tasks that are insensitive to syntactic and semantic variables, relying only on an independently operating lexical 'parser'. We can also find tasks that are sensitive to syntactic but not semantic factors.<sup>48</sup> If the grammar incorporated all-purpose strategies, sensitive to each grammatical level simultaneously, then we would expect syntactic, semantic, and lexical effects to be inseparable. The information would have been encoded into general rule types and no independent access would be possible.

There is another interpretation to 'separation of components' that seems to be at least compatible with the Tyler and Marslen-Wilson data. Under this interpretation, we retain the idea that the grammar incorporated in the parser consists of independent rule systems. The difference is that rather than compiling a complete analysis of the string from the output representations constructed by the various subcomponents, we would construct this analysis directly by allowing the subcomponents (and perhaps some non-linguistic 'real-world' knowledge systems) to interact on-line to build a single representation. As mentioned before however, this position, because it distinguishes separate rule types, is perfectly compatible with the autonomy of syntax thesis as it is specified in TGG. Thus, we can still claim that a processing model of this type respects the *logical* organization of components implied by a TGG, even though this logical organization does not correspond in any transparent way to the *flowchart* description of the processing model.

#### 4. Type Transparency and the theory of grammar

In the course of this paper we have examined two cases—the DTC and the Tyler and Marslen-Wilson experiments—where the Type Transparency Hypothesis played a crucial role in the initial evaluation of transformational generative grammar as a central component of a model of language use. Recall that the Type Transparency Hypothesis amounts to the assumption that the substantive elements and derivations implied by a grammar should be taken as specifying the exact composition and organization of parsing algorithms that incorporate that grammar. In this view, a particular grammar is 'realized' in a parsing model just in case derivations of sentences proposed by a grammar mimic exactly the sequence of computational steps that the

---

<sup>48</sup>See Forster, 1979, pp. 39–41 for details.



parser makes in building structural descriptions for the same sentences. This thesis was of interest to parsing theorists because it constrains the class of possible parsers to a small set; i.e., to that set that can be used to parse natural language according to the structural descriptions and rules of TG written in very much the same format as they are in the grammar. In Section Two of this paper we showed that it is possible to weaken this notion of 'realization' from one of exact mimicry, and still obtain perspicuous grammar-parser combinations that reproduce known psycholinguistic results. For instance, in our concrete discussion of extended lexical grammar, we showed that Bresnan has employed a less than direct notion of 'realization' to make extended lexical grammars compatible with both certain computational assumptions (a serial parsing model valuing memory retrieval over 'brute' recomputation) and certain psycholinguistic results (active/passive processing distinctions). We also showed that a transformational grammar could exploit the same computational assumptions. The explanatory underpinning for the psycholinguistic distinctions hinges on our ability to provide formal criteria to distinguish rule types as different in the *grammar*. This grammatical difference is then preserved by the parser as a difference in algorithmic realization, and hence an explanation for a measurable difference in processing time. The class of possible parsers is still constrained to those that preserve this grammatical rule/parsing algorithm type-type identity. Note that the parser need not mimic the 'rules of the grammar' exactly, so long as the distinctions made by the grammar are *preserved*: in this approach, there might be a variety of representational formats and computational organizations (e.g., Bresnan's memory retrieval scheme for lexical templates) that could all 'realize' the same grammar.<sup>49</sup>

In the particular cases examined above the formal characterization of the processing algorithm was rather close to the formal characterization of the associated grammar. For instance, in the case of extended lexical grammar

---

<sup>49</sup>As pointed out earlier, the parser would still be licensed to make *more* distinctions than the grammar, if necessary.

As Matthews (1979) points out, it is not at all apparent that the parser should have to characterize as acceptable all and only the grammatical strings of a language. Indeed, as is well known, the set of acceptable sentences can be more limited than the set of grammatical sentences, since the analysis of some sentences may exceed the memory capacity of a (finite) parsing algorithm. (The standard case in point is multiple center-embedded sentences, e.g., *the rat the dog ate died*.) This again suggests that the parser is a restricted version of the grammar. On the other hand, distinctions between grammaticality and acceptability may cut the other way: we may find ungrammatical sentences that are acceptable (see Hornstein and Weinberg [1981] for an example). Taken together, these results suggest that the difference between grammar and parser may yield different predictions about the analysis of sentences of a given language in a small subset of cases. Characterizing the exact relationship between grammaticality and acceptability is a challenging task that we leave for future research.

the parser uses the same structural descriptions as the grammar, and, for a subset of cases (*wh* movement rules) the rule format specified by the grammar is preserved in the parsing algorithm. Moreover in all cases, even if the actual rule used in the grammar is not used by the parser, the different representational levels are related in the way specified by the grammar. For example, the interpretation of sentences in an ELT-based parser includes a mapping between phrase structure and functional structure. In the TG-based model, the interpretation is a mapping between Deep Structure and Surface Structure (or an interpretation from Annotated Surface Structure in the EST version) that is executed by movement (ST version) or by binding (EST version). These approaches are direct, transparent, embeddings of grammars as parsing models. And it is interesting to have discovered that there is currently no psycholinguistic evidence that prevents us from adopting this fairly direct kind of mapping.

However, transparency is not a *necessary* property of a parsing model. If future experiments show that this direct mapping is untenable, then researchers interested in constructing a theory of language use should still be interested in the theory of linguistic competence, to the degree to which we can use this theory to constrain the class of possible parsers.

In this section we show that there is a continuum of more to less direct realizations of a grammar as a parser. There is not just an 'all or none' choice between a grammar embedded directly as a computational model (the DTC model) and a total decoupling between grammatical rules and computational rules (with the structural descriptions of the grammar computed by some totally unrelated 'heuristic strategies', the Fodor, Bever and Garrett [1974] conclusion). By using a relation among a class of grammars known as *covering*, one can demonstrate that a parser may be able to exploit the rules of a grammar non-transparently. Even under these conditions the class of possible parsers may be restricted by grammatical theory.

To see that the relation between grammar and parser could be less than transparent, we shall examine in some detail a measure of similarity developed in the context of research on programming languages, that of *grammatical cover*.<sup>50</sup> Intuitively, one grammar is said to *cover* another if the first grammar can be used to (easily) recover all the parses (structural descriptions, labelled bracketings) that the second grammar assigns to input sentences. This being so, it should be plain that the first grammar can be used instead of the second grammar itself to parse sentences of the language

<sup>50</sup>The notion of grammatical cover was used informally for several years before being formalized by Reynolds (1968) and Gray and Harrison (1969). As will be made clear below, the insight that covering grammars could be of value in developing parsing models for natural languages is not a new one; in fact, the idea is found in, e.g., Kuno (1965).

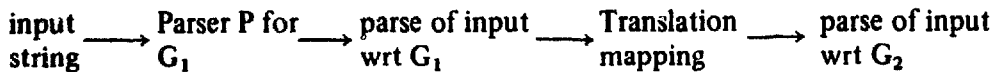
generated by the second grammar. More importantly, the cover relation provides a rich stock of cases where two grammars can generate trees that do not necessarily look very much alike, and yet one grammar can serve as the 'true' competence grammar for a language (because it generates the proper structural descriptions) while the other can be used to efficiently parse the language (because of certain special structural characteristics of the trees it generates). In short, this approach allows us to hold the structural descriptions of a grammar fixed, and then to consider variations in parsing methods. The theory of grammar will limit the class of possible parsers to just those that cover the original competence grammar. This is possibly a strong limitation, hence of potential interest to parsing theory.

Such cases provide real examples of the existence of non-transparent ways to incorporate grammars into models of language use. Having settled the question of the existence of such grammar-parser pairs, the next question to ask is whether there is any advantage to explicitly separating out the levels of grammar and parser in this manner. In the remainder of this section we shall advance some reasons as to why it is advantageous to do so. In brief, it is suggested that one virtue of an explicit decomposition into grammar, parser (and, perhaps, implementation) is to permit a modular attack on the explanation of a complex information processing system, namely, the language faculty; in addition, there seem to be explanations of inherent psychological interest (like accounts of language acquisition) that make crucial reference to a separate level of grammatical representation.

Returning now to the notion of grammatical cover, it is easy to show that there are well-known (but degenerate) examples of pairs of grammars that cover each other. Consider a grammar that is *strongly equivalent* to another grammar. By the usual definition of strong equivalence, this means that the first grammar generates *exactly* the same set of structural descriptions as the second, and hence, *a fortiori*, covers the second grammar. (Recall, for contrast, that one grammar is *weakly equivalent* to another grammar if it can generate the same set of terminal strings—roughly, the same sentences—as the second grammar; in this case it need not generate the same structural descriptions at all).

If this were the only example of grammatical covering, then the cover relation would collapse to the usual notion of strong equivalence. But it is also true that one grammar can cover another under far less stringent conditions of similarity. Informally, one grammar  $G_1$  covers another grammar  $G_2$  if (1) both generate the same language [ $L(G_1) = L(G_2)$ , that is, the grammars are *weakly equivalent*] and (2) we can find the parses (structural descriptions) that  $G_2$  assigns to sentences by parsing the sentences using  $G_1$  and then applying a 'simple' (easily computed) mapping to the resulting output. (We

shall be more precise shortly about what is meant by 'simple'). Note that these two grammars need not be strongly equivalent, and yet the first can still be used to parse sentences generated by the second; for the purposes of *parsing*, such grammars are equivalent.<sup>51</sup> The picture then is roughly the following:



Given some 'correct' grammar for a language (i.e., a grammar that generates the right structural descriptions), why would one want to parse sentences using a different, but covering, grammar rather than the correct grammar itself? The reason is that the covering grammar may be more 'suitable' for parsing, along any one of a number of dimensions (efficiency of processing in terms of time or space use, perspicuity, compatibility with fixed hardware, and so forth). In this view, the 'correct' grammar provides the right structural descriptions for rules of semantic interpretation, whereas the covering grammar provides the right format for algorithmic instantiation. (These two grammars might be the same, in which case we obtain strict transparency, a one-to-one grammar-parser relation). The cover mapping takes trees generated by the covering grammar (used for parsing) into trees generated by the 'correct' grammar (used for semantic interpretation). Thus, if one grammar covers another, then whatever the rules of semantic interpretation (let us say they pair parse trees with 'meanings'), either grammar can be used to pair exactly the same input strings and meanings. Let us now return to exactly what is meant by a *simple* mapping in the definition of grammatical cover. As the above figure shows, the mapping from parse trees to parse trees must be tightly restricted, or else any grammar could be made to cover any other grammar. The usual definition of 'simple' made in the formal language theory literature (Aho and Ullman, 1972, page 275) is that of *string homomorphism*. That is, if the parse of a sentence with respect to a grammar  $G_1$  is represented as a string of *numbers* (corresponding to the rules that were applied to generate the sentence, under some arbitrary numbering of the rules of the grammar and some canonical derivation sequence), then the translation mapping that carries this string of numbers (a parse) to a new string (corresponding to another parse) must be a homomorphism (under concatenation). Note crucially that the homomorphic recovery can proceed on-line, incrementally and left-to-right as the parse

<sup>51</sup> By 'parsing' we simply mean 'recover structural descriptions'; we have abstracted away from a consideration of time and space resource use. It might well turn out (as will be shown below) that two grammars equivalent with respect to parsing could be radically different with respect to their computational efficiency.

proceeds, and that if the mapping is fixed in advance the computation can be done quite rapidly;<sup>52</sup> what this means is that the desired parse tree can be recovered with little computational loss.<sup>53</sup>

In summary then, covering grammars provide an abundant source of examples illustrating that the grammar used by the parser or sentence processor need not *directly* generate structural descriptions isomorphic to those specified by the grammar of the competence theory. Indeed, the sets of structural descriptions directly constructed by the parser and generated by the grammar can look quite different, and yet the parser can still faithfully mirror the competence grammar by incorporating a cover homomorphism that recovers the proper structural descriptions as required. Furthermore, this approach has proved of actual value in constructing parsers for natural language (see Kuno, 1966), and in developing optimal processors for programming languages (see Bochmann, 1979; Brosgol, 1974; Hammer, 1975). As far as parsing is concerned then, both the theory and

<sup>52</sup>In fact, it will take only a fixed number of steps per individual rule used in the derivation.

<sup>53</sup>This result is a significant one for those who study models of language use, because it illustrates how a parser (e.g., incorporating a covering grammar plus cover homomorphism) can be non-transparently related to the grammar it instantiates. This is not just a theoretical possibility. In fact, the study of 'transformations' from one grammar into another that preserve recoverability of structural descriptions and at the same time improve amenability to parsing is a long-standing area of research in formal language theory and in programming language analysis. (See Foster, 1968; Hammer, 1975; Kuno, 1966; Kurki-Sunio, 1966).

For example, consider Kuno's Harvard Syntactic Analyzer (1966). As Nijholt observes (1980), Kuno's basic aim was exactly that of finding grammatical transformations that ensured parsing efficiency while at the same time preserving recoverability of structural descriptions. Kuno even includes a diagram just like the one above as the framework for his research. More specifically, Kuno attempted to find mappings from arbitrary context-free grammars (for natural languages) to grammars that were efficiently handled by a particular (top-down) parser he had constructed, so-called *non left-recursive context-free grammars*. A non-left recursive grammar contains no productions of the form,  $A \Rightarrow \dots \Rightarrow Ab$  (Where A derives itself by one or more rules).

The important point is that top-down parsers can be *guaranteed* to work only if they incorporate non-left recursive grammars. Kuno compiled a list of ways to change a context-free grammar with left-recursive productions into a non-left recursive one, and hence one amenable for top-down parsing; in so doing he obtained a grammar that (right)-covered the original.

In particular, Nijholt has shown that every context-free grammar (without single 'erasing' productions of the form  $A \Rightarrow \text{empty}$ ) can be right-covered by a grammar in Greibach-normal form (a canonical representation of a context-free grammar where every production is of the form  $A \Rightarrow bB$ , where  $b$  = a terminal symbol and  $B$  = a string of non-terminals). Note that Greibach normal form is non-left recursive. Nijholt's result thus provides one way to convert 'almost any' context-free grammar into a form that can be used by a top-down parser.

Kuno also discovered a non-left recursive form for context-free grammars, but could not find a covering grammar in Greibach normal form. Nijholt's result thus answers in the affirmative a problem posed by Griffiths and Petrick: "To date (1965 rcb/asw) no effective procedure for relating the structural descriptions of standard normal form (Greibach normal form rcb/asw) grammars to the context-free grammars from which they were constructed has been found" (Griffiths and Petrick, 1965).

A covering grammar provides an easily computed mapping between the structural descriptions of a Greibach grammar and the SD's of its associated grammar or origin.

practice of parser design have made considerable use of the concept of a *non-transparent* relation between grammar and parser, that of grammatical cover.<sup>54</sup>

One might however ask just why one needs to specify a level of purely grammatical characterization at all. Why not simply dispense with grammar and just look at parsing algorithms directly, incorporating notions of computation from the start? Why should theorists interested in language use be concerned with a level of grammatical description at all? Indeed, some have suggested abandoning grammar as the proper subject matter for psycholinguistic investigation:

The proper task for the psycholinguist is not, at the moment, to determine the relationships between linguistic theory and psychological process, but to try to acquire the kind of psychological processing data which will allow the construction of a genuinely psychological theory of sentence recognition (Tyler, 1980, page 58).

Given the discussion in Sections Two, Three, and above, the problems with jettisoning grammar altogether should be obvious. First of all, we have seen that one can use the theory of competence to constrain the class of possible parsers. The choice of one parsing algorithm over another depends on a great many factors, including what structural descriptions the parser is supposed

---

<sup>54</sup>Recalling our discussion of Watt (1970) in Appendix 1, we observed there that Watt takes the fact that short passives require no more parsing time than long passives as evidence that one must abandon a model that directly realizes a TG. Our discussion so far indicates that there are at least two alternatives to this move. First, even if TG claimed that short passives are derived by deletion from long ones (which it currently does not), we could still embed a TG into a non-serial parsing model. One might assume that the parser generates a postverbal agentive 'by' phrase with an empty object immediately upon the recognition of the passive morphology. This would predict that comprehension time for short and long passives should be about the same. The possibility of concurrency undermines Watt's implicit assumption that 'economy of derivation' in terms of time cannot be preserved in a theory that relies heavily on computation to generate or analyze strings. Second, Watt seems to assume that the realization of a 'competence grammar' in a model of language use is an all or nothing proposition. In his terms, a speaker's 'mental grammar' consists of an 'abstract performative grammar' (APG) and a 'competence grammar'. The APG is the grammar actually used in parsing. The principles that govern its construction ensure an analysis of sentences that involve a minimum of computation and thus, according to Watt, result in a rule system that predicts the experimentally observed time complexity results: "the APG puts a premium on economy of derivation and so balks at incorporating some of these generalizations (generalizations motivated by linguistic theory, e.g., a deletion rule for deriving short passives from long passives, rcb/asw)" (1970, page 181).

Watt, like Fodor, Bever and Garrett (see Appendix 1) realizes that the APG grammar will miss much that the speaker undoubtedly knows about his language. Therefore, Watt concludes that the APG grammar is to be supplemented by an 'archival competence faculty', basically the grammar of linguistic theory. This grammar may or may not be used in on-line processing; it will be where it accords with processing complexity data.

In brief, Watt feels that either the 'parsing grammar' is isomorphic to 'competence grammar', or the APG is based on principles radically different from the competence grammar. As mentioned above, we would like to propose a more refined hierarchy of possibilities.

to compute and what computational organization is presumed to carry out the computation. As we have tried to make clear, there is very little that is known for certain about just what computational organization is actually used for human language processing—we cannot even specify its basic structure with any degree of certainty. Moreover, though we have much firmer evidence about the correct characterization of linguistic knowledge in the abstract, even that characterization (a theory of grammar) is presumably far from the millennial theory of grammar. Consequently, any complete theory of language use must of necessity be built upon a doubly incomplete knowledge about the language faculty and the human computational machinery, and, as a result, one is bound to have difficulty predicting psycholinguistic data of any complexity.

Why does separating out the explanatory levels as grammar and algorithmic realization help? Assuming that we have kept these levels distinct, then it becomes easier to determine just what is contributing to discrepancy between theory and surface facts. For instance, if levels are kept distinct, then one is able to hold the grammar constant and vary machine architectures to explore the possibility of a good fit between psycholinguistic evidence and model. (This was our approach in Section Two). Suppose these results came to naught. We can then try to co-vary machine architecture and covering mappings, still seeking model and data compatibility. If this fails, one could still try different grammars. In short, modularity of explanation permits a corresponding modularity of scientific investigation. For a complex information processing system like the language faculty, this may be the investigative method of choice; in fact, this has been the research strategy adopted by D. Marr and others in their study of early visual processing, a strategy that has paid off with impressive results:

... in a system that solves an information processing problem, we may distinguish four important levels of description... At the lowest, there is basic component and circuit analysis—how do transistors (or neurons), diodes (or synapses) work? The second level is the study of particular mechanisms: adders, multipliers, and memories, these being assemblies made from basic components. The third level is that of the algorithm, the scheme for a computation; and the top level contains the theory of the computation. ... [T]ake the case of Fourier analysis. Here the computational theory of the Fourier transform—the decomposition of an arbitrary mathematical curve into a sum of sine waves of differing frequencies—is well understood, and is expressed independently of the particular way in which it might be computed. One level down, there are several algorithms for computing a Fourier transform, among them the so-called Fast Fourier Transform (FFT), which comprises a sequence of mathematical operations, and the so-called spatial algorithm, a single, global operation that is based on the mechanisms of laser optics. All such algorithms produce the same result, so the choice of which one to use

depends upon the particular mechanisms that are available. If one has fast digital memory, adders, and multipliers, then one will use the FFT, and if one has a laser and photographic plates, one will use an 'optical' method (Marr and Poggio, 1977).

In contrast, in a system where the theory of grammar, parser, and machine are collapsed together one cannot decouple these levels of description so as to settle, e.g., questions of *what* is being computed as distinct from *how* it is computed. If discrepancies between external facts and theory arise in this sort of approach, one must either reformulate the entire theory, or backtrack and attempt to extract properties invariant with respect to algorithmic or machine instantiation. Given our current limited understanding, one would expect many such discrepancies between data and model, and so the 'constant reformulation' strategy seems fruitless. The other route amounts to what Marr suggested in the first place: to separate out the levels of abstract theory, algorithm, and implementation so that results can be at least roughly carved out at each level in a more independent fashion.

Besides the tactical advantage accruing to a modular approach, there seem to be basic questions of psychological interest whose answers crucially refer to a level of grammatical description. For example, most questions of 'language learning' appear to be most perspicuously answered by reference to grammars, and there even seem to be examples where the theory of language use and language acquisition make contact.<sup>55</sup> Let us take one example that illustrates how grammars are implicated in theories of language acquisition.

Assume (counterfactually, in the opinion of the authors) that the theory of extended lexical grammar provides the optimal grammar for English and that the processing model outlined in, e.g., Bresnan (1978) is in fact the one embodied by the language faculty.

Now assume that at some stage of acquisition one encounters a new verb, say, *to disambiguate*, in the following context:

The context disambiguated the meaning of the sentence.

Given the framework of ELT, this verb would be stored in the lexicon with its functional representation: *disambiguate*: [NP<sub>1</sub>—NP<sub>2</sub>]. It seems clear however that a native speaker, even without having met an exemplar, could easily recognize the passive counterpart to the sentence above:

The meaning was disambiguated by the context of the sentence.

---

<sup>55</sup> See Berwick (1979, 1980, 1982 [Reference note 2]) for a model that explicitly connects a model of language use (the Marcus parser) to a model of language acquisition.



ELT would claim that this is because the lexicon could also associate a passive template with the lexical entry for *disambiguate*, once the active form had been built. But on what basis? Presumably, there is some sort of active-passive relation, perhaps captured by an explicit rule (a lexical redundancy rule; see Section 1.1 above) that can construct the passive entry once the active entry is built. However, note that this active-passive relation is *crucially* not part of the parsing algorithm itself (its effects can be encoded into the lexicon independently of the parsing algorithm), but is statable only at a level of grammatical description. Therefore, in order to show how new lexical forms are integrated into the parser—a problem presumably of obvious interest to psychologists—one must crucially refer to a level of grammatical representation. Many more such examples drawn from the study of acquisition could be cited, but they would take us far beyond the scope of this paper.

Three major points emerge from this investigation of the connection between grammars, parsers, and machine implementations. First, given our limited knowledge about the range of possible machine implementations, it would seem ill-advised to directly specify a theory of parsing for natural language without first having a good theory of grammar—the knowledge the parser is to work with. Second, the theory of grammar can go a long way to delimit the class of possible parsing algorithms (since it specifies the function to be computed by the parser). Finally, some questions relevant to the theory of language use are apparently answerable only by direct reference to the level of grammatical representation. Each of these conclusions points to a single moral: it seems that the development of an adequate theory of language use will depend on a firm characterization of linguistic knowledge, a grammar. One cannot build a theory of language use directly; the theory of language use will emerge out of a theory of competence, a theory of algorithms, a theory of implementation, and a theory of the proper mapping between these explanatory levels.

## Appendix I. The evaluation of other DTC experiments

There are several other experiments that purport to falsify the DTC. We treat them separately because they are of a very different character than the Slobin results.

We argued above that the Slobin results might suggest either (1) reorganizing the grammar so that it may be incorporated into the parser (the suggestion advanced by Bresnan), or (2) reorganizing the parser (which we have suggested). In both cases, one of the basic assumptions of the DTC was revised.

In contrast, the experiments we consider in this Appendix seem to show either that the details of the grammar (Standard Theory) assumed by the DTC are incorrect or that the distinctions among rules that have a natural interpretation within the Standard Theory (or Extended Standard Theory) may be reflected in the parser. In other words, these experiments may suggest minor modifications in the DTC, but leave its basic theses untouched.<sup>56</sup>

For example, Fodor and Garrett (1967) performed an experiment contrasting sentences whose Noun Phrases were modified by a series of prenominal adjectives with sentences containing only bare Noun Phrases. They assumed a theory that derived prenominal adjectives from relative clauses by a reduction and preposing operation called *whiz-deletion*.<sup>57</sup> Under this analysis, a phrase like (1) was derived from (2) by first deleting the *wh* and verb sequence and then preposing the adjective to prenominal position:

- (1) The red book  
from
- (2) The book which is red.

Assuming that *whiz* deletion is a transformation, constructions with prenominal adjectives should take more total time to parse than those containing basic Noun Phrases. Fodor and Garrett found, however, that prenominal modification produced no complexity effect: "The sentences with adjectives exhibited no tendency to inhibit subjects' accuracy on the paraphrase task" (Fodor, Bever and Garrett, 1974, page 325). Thus, their results tend to indicate that prenominal adjectives are not derived using transformations.

This same conclusion was reached independently by Williams (1975). Williams provides a host of purely syntactic arguments against the *whiz* deletion analysis. He offers an alternative theory where prenominal adjectives are simply generated in place. He then shows how this analysis fits in with the kind of TG proposed by Chomsky (1970). Assuming Williams's analysis, we may predict the complexity results of Fodor and Garrett while staying entirely within the framework of the Standard Theory.

Another purported counterexample to the DTC is reported in Watt (1970). Watt presupposes a theory (advanced by many linguists in the 1960's) whereby 'short passives' (such as *John was hit*) are derived from long passives (*John was hit by Fred*) via a deletion of the agentive 'by' phrase.

<sup>56</sup>This possibility is also mentioned in Valian (1979, page 6) and in Cooper and Walker (1979).

<sup>57</sup>See Ross (1967). Ross's motivation for this analysis was the assumption that phrases that have the same meaning should have the same deep structures (a basic assumption of the branch of transformational grammar called generative semantics). Since phrases like *the red door* and *the door which is red* have the same meaning, these phrases were presumed to be derived from the same underlying deep structure source. See Chomsky (1970) for theoretical and empirical arguments against this assumption.

Watt (following Fodor and Garrett [1967]) claims that short passives take no longer to parse than their longer counterparts, a problem for the DTC on the assumption that the deletion operation involved in generating short passives adds time complexity to the analysis of these sentences. However, current linguistic theory generates short passives directly; no deletion of a 'by' phrase is required. (See Bresnan, 1972 [Reference note 3] and Chomsky, 1976 for evidence). Given this, even a TG embedded directly in a serial processing model would predict the data that Watt cites.

The next set of experiments that have been taken to impugn the direct embedding of a TG in a processing model deal with so-called *particle* and *adverbial* movements. In unpublished experiments (reported in Fodor, Bever and Garrett [1974]), Bever, Fodor, Garrett and Mehler compared the following sentences:

- 1(a) John phoned up the girl.
- 1(b) John phoned the girl up.

Bever and Mehler (see Bever [1968]) compared sentences like 2(a) and 2(b):

- 2(a) Slowly the operator looked the number up.
- 2(b) The operator slowly looked the number up.

At one time both of the (b) examples were presumed to be derived transformationally from the corresponding (a) examples. 1(b) was derived from 1(a) by particle movement (see Emonds, 1976), and 2(b) from 2(a) by adverb preposing (see Emonds, 1976 and Keyser, 1968). Given the assumptions of the DTC we would predict that the (b) examples would be more complex than the (a) examples. However, no such complexity effects were uncovered.

At the time when the relevant complexity experiments were done, people believed that transformations were the only possible structure changing rules. Subsequent studies have shown that this traditional conception of the transformational component was too broad in that it lumped together rules that had very different properties.<sup>58</sup> In the case at hand, the (b) examples clearly mean exactly the same thing as their (a) counterparts. In general the application of rules such as particle movement and adverbial preposing does not affect the semantic interpretation of these sentences at all. There is no *semantic* reason to relate an adverb or a particle to its purported deep structure position (as opposed to passive; see the discussion in the main text above). Secondly, these rules do not apply successive cyclically. (Roughly, they cannot move a constituent out of the clause from which it originates). Consider, for example:

<sup>58</sup>See Emonds (1972, 1976) and Chomsky and Lasnik (1976) for comprehensive discussion of this issue.

- 3(a) I said that the operator quickly dialed the number.  
 3(b) I said quickly that the operator (empty) dialed the number.

Although 3(b) is grammatical it cannot be derived from 3(a) because in 3(a) the adverb modifies the verb *to dial* while in 3(b) it modifies *said* and has no relation at all to the lower verb. It seems reasonable to assume then that the adverbs in these sentences simply originate in the clauses where they are found in surface structure. Adverb movement is only clause internal.

In the case of particle movement, the particle cannot even be moved from the Verb Phrase from which it originated. Compare:

- 4(a) I called the operator up yesterday.  
 4(b) \*I called the operator yesterday up.

4(b) is ungrammatical because since *yesterday* is a sentence adverb its derivation requires movement of the particle *up* out of the Verb Phrase and into a slot directly under the S. This is impossible if we assume (see Emonds, 1976) that the domain of particle movement is strictly 'local' (internal to the Verb Phrase).

These properties contrast sharply with those of the so-called passive rule (and with the properties of Noun Phrase and *wh*-movement in general). Consider:

- 5(a) John was believed to have been seen by Bill.

To derive this sentence we start with a deep structure:

- 5(b) [ ] believed [ ] to have been seen John by Bill.

Next we apply the passive rule:

- 5(c) [ ] was believed John to have been seen by Bill.

Finally we re-apply this rule (successive cyclically), obtaining 5(a):

- 5(d) John was believed to have been seen by Bill.

It is clear (in contrast to the particle and adverb movement cases) that we must relate *John* back to its deep structure position because the proper interpretation of this sentence requires that *John* be construed as the object of *be seen*.

Because rules like particle movement and adverb placement have properties that are so different from those of rules like passive, it has been proposed (see Chomsky and Lasnik, 1977 and Dresher and Hornstein, 1977) that these rules be classified as stylistic rules and be separated from transformations.

Clearly under such a proposal one could still retain the DTC by claiming that transformations but not stylistic rules are actually computed during

sentence processing and therefore add to its complexity. Since particle movement and adverb placement do not influence semantics, we could perhaps claim that 'de-transforming' operations need not apply in these cases; the particles or adverbs are simply left in place, as they appear in the input string. This would accord with the DTC hypothesis that only structure relevant to semantic interpretation need be recovered during sentence processing.<sup>59</sup>

The important point is that even the Standard Theory provides the distinctions needed to make the DTC compatible with the cited psycholinguistic complexity results.

## References

- Aho, A., and Ullman, J. (1972) *The Theory of Parsing, Translation, and Compiling*. Englewood Cliffs, NJ, Prentice-Hall.
- Anderson, J. (1978) Arguments concerning representations for mental imagery, *Psychol. Rev.*, 85, 249–277.
- Anderson, J. (1979) Further arguments concerning representations for mental imagery: A response to Hayes-Roth and Pylyshyn. *Psychol. Rev.*, 86, 395–406.
- Armstrong, (1979) Recursive solution to the equations of motion of an N-link manipulator. *Proc. 5th World Congress on Theory of Machines and Mechanisms*, Vol. 2, 1343–1346.
- Berwick, R. (1979) Learning structural descriptions of grammar rules from examples. Pp. 56–58 in *Proc. 7th Internat. Joint Conf. Artificial Intelligence*, Tokyo.
- Berwick, R. (1980) Computational analogues of constraints on grammars: a model of syntactic acquisition. Pp. 49–53 in *Proc. 18th An. Meeting Assoc. Computational Linguistics*, Philadelphia, PA.
- Berwick, R. (1981a) Computational complexity and lexical functional grammar. Pp. 7–12 in *Proc. 19th An. Meeting Assoc. Computational Linguistics*, Stanford, CA.
- Bever, T. (1968) A survey of some recent work in psycholinguistics. In W. J. Plath (ed.), *Specification and Utilization of a Transformational Grammar*. Scientific Report #3. IBM Thomas J. Watson Research Center, Yorktown Heights.
- Bochmann, G. (1979) Semantic equivalence of covering attribute grammars. *Int. J. Comp. Inf. Sci.*, 8, 523–540.
- Borođin, A. (1977) On relating time and space to size and depth. *SIAM J. Comp.*, 6(4), 733–744.
- Bresnan, J. (1978) A realistic transformational grammar. In J. Bresnan, M. Halle, and G. Miller, (eds.), *Linguistic Theory and Psychological Reality*. Cambridge, MA, MIT Press.
- Bresnan, J. (In press) *The Mental Representation of Grammatical Relations*. Cambridge, MA, MIT Press.
- Brosgol, B. (1974) *Deterministic Translation Grammars*. PhD. Dissertation, Harvard University, Aiken Computation Lab TR-3-74.
- Chomsky, N. (1965) *Aspects of the Theory of Syntax*. Cambridge, MA, MIT Press.
- Chomsky, N. (1968) *Language and Mind*. New York, Harcourt, Brace, Jovanovich.
- Chomsky, N. (1970) Remarks on nominalization. In R. Jacobs and P. Rosenbaum (eds.), *Readings in Transformational Grammar*. Waltham, MA, Blaisdell.

<sup>59</sup>There are other alternatives to deal with 'local' movement rules in a parsing model; see Marcus (1980) for some examples.

- Chomsky, N. (1976) Conditions on rules of grammar. *Ling. Anal.*, 2, 303–351.
- Chomsky, N. (1977) On *wh*-movement. In P. Culicover, T. Wasow, and A. Akmajian (eds.), *Formal Syntax*. New York, Academic Press.
- Chomsky, N. (1979) *Language and Responsibility*. New York, Pantheon.
- Chomsky, N. (1980) *Rules and Representations*. New York, Columbia University Press.
- Chomsky, N. (1981) Lectures on Government and Binding, Dordrecht, Foris.
- Chomsky, N., and Lasnik, H. (1977) Filters and Control. *Ling. Inq.*, 8(4), 425–504.
- Cook, S. A. (1980) Towards a complexity theory of synchronous parallel computation. *University of Toronto Computer Science Report*, #141/80. Toronto, Canada.
- Cooper, W., and Paccia-Cooper, J. (1980) *Syntax and Speech*. Cambridge, MA, Harvard University Press.
- Cooper, W. E., and Walker, E. C. T. (1979) *Sentence Processing: Psycholinguistic Studies Presented to Merrill Garrett*. Hillsdale, NJ, Lawrence Erlbaum.
- Dresher, E., and Hornstein, N. (1976) On some supposed contributions of AI to the scientific study of language. *Cog.*, 4, 321–398.
- Dresher, E., and Hornstein, N. (1979) Trace theory and NP movement rules. *Ling. Inq.*, 10(1), 65–82.
- Emonds, J. (1970) *Root and Structure Preserving Transformations*. PhD dissertation, MIT Department of Linguistics. (Published in revised form as Emonds [1976]).
- Emonds, J. (1972) A reformulation of certain syntactic transformations. In S. Peters (ed.), *Goals of Linguistic Theory*. Englewood Cliffs, NJ, Prentice-Hall.
- Emonds, J. (1976) *A Transformational Approach to English Syntax*, New York, Academic Press.
- Fodor, J., and Garrett, M. (1967) Some syntactic determinants of sentential complexity, *Percept. Psychophys.*, 2, 289–296.
- Fodor, J., Bever, T., and Garrett, M. (1974) *The Psychology of Language*. New York, McGraw-Hill.
- Forster, K. (1979) Levels of processing and the structure of the language processor. In W. E. Cooper and E. C. T. Walker, (Eds.), *Sentence Processing: Psycholinguistic Studies Presented to Merrill Garrett*. Hillsdale, NJ, Lawrence Erlbaum.
- Forster, K., and Olbrei, I. (1973) Semantic heuristics and syntactic analysis. *Cog.*, 2, 319–348.
- Foster, J. (1968) A syntax improving program. *Comput. J.*, 11, 31–34.
- Freidin, R. (1978) Cyclicity and the theory of grammar. *Ling. Inq.*, 9(4), 519–550.
- Goldschlager, L. (1978) A unified approach to models of synchronous parallel computation. Pp. 89–94 in *Proc. 11th ACM Symp. Theory of Computing*.
- Gough, D. (1965) The verification of sentences. *The J. verb. Learn. verb. Behav.*, 5, 107–111.
- Gray, J., and Harrison, M. (1969) Single pass precedence analysis. Pp. 106–117 in *IEEE Conf. Record 10th An. Symp. on Switching and Automata Theory*.
- Greibach, S. (1965) A new normal form theorem for context-free grammars. *J. Assoc. Comput. Mach.*, 12, 42–52.
- Griffiths, T., and Petrick, S. (1965) On the relative efficiencies of context-free grammar recognizers. *Communic. Assoc. Comput. Mach.*, 8, 289–300.
- Grimson, E., and Marr, D. (1979) A computer implementation of a theory of human stereo vision. Pp. 41–47 in *Proc. Image Understanding Workshop*. Palo Alto, CA.
- Hammer, M. (1975) *A New Grammatical Transformation into Deterministic Top-down Form*, Cambridge, MA, MIT Project MAC TR-119.
- Hayes-Roth, F. (1979) Distinguishing theories of representation: a critique of Anderson's "Arguments concerning mental imagery". *Psychol. Rev.*, 86, 376–382.
- Horn, B., and Raibert, M. (1978) Configuration space control. *The Industrial Robot*, 5, 69–73.
- Hornstein, N., and Weinberg, A. (1981) Case theory and preposition stranding. *Ling. Inq.*, 12(1), 55–92.

- Kaplan, R., and Bresnan, J. (1981) Lexical functional grammar: A formal system for grammatical representation. MIT Center for Cognitive Science Occasional Paper #13, to appear in J. Bresnan (ed.), *The Mental Representation of Grammatical Relations*. Cambridge, MA, MIT Press.
- Keyser, S. J. (1968) Review of Sven Jacobson, adverbial positions in English, *Lang.*, 44, 357–374.
- Kintsch, W. (1974) *The Representation of Meaning in Memory*. Hillsdale, NJ, Lawrence Erlbaum.
- Kung, H. (1976) Synchronized and asynchronous parallel algorithms for multiprocessors. In J. Traub, (ed.), *Algorithms and Complexity: New Directions and Recent Research*. New York, Academic Press.
- Kuno, S. (1965) The augmented predictive analyzer for context-free languages—its relative efficiency, *Communic. Assoc. Comput. Mach.*, 9, 810–823.
- Kurki-Suonio, R. (1966) On top-to-bottom recognition and left recursion, *Communic. Assoc. for Comput. Mach.*, 9, 527–528.
- Levelt, W. (1974) *Formal Grammars in Linguistics and Psycholinguistics*. The Hague, Mouton.
- Luh, J., Walker, M., and Paul, R. (1979) On-line computational scheme for mechanical manipulators. *2nd IFIP Symp. Inf. Control Problems in Manufacturing Technology*, Stuttgart, Germany.
- Machtey, M., and Young, P. (1978) *An Introduction to the General Theory of Algorithms*. New York, North-Holland Press.
- Marcus, M. (1980) *A Theory of Syntactic Recognition for Natural Language*. Cambridge, MA, MIT Press.
- Marr, D., and Hildreth, E. (1980) The smallest channel in early human vision, *J. Optic. Soc. Amer.*, 70(7), 868–870.
- Marr, D., and Poggio, T. (1977) From understanding computation to understanding neural circuitry. *Neurosc. Res. Prog. Bull.*, 15(3), 470–488.
- Marr, D., and Poggio, T. (1979) A theory of human stereo vision, *Proc. Roy. Soc. London B.*, 204, 301–328.
- Matthews, R. (1979) Are the grammatical sentences of a natural language a recursive set?, *Synthese*, 40, 209–224.
- Mehler, J. (1963) Some effects of grammatical transformations on the recall of English sentences, *J. verb. Learn. verb. Behav.*, 6, 335–338.
- Meyer, A., and Shamos, M. (1977) Time and space. In A. K. Jones, (ed.), *Perspectives on Computer Science*, New York, Academic Press.
- Miller, G., and Chomsky, N. (1963) Finitary models of language users. In R. Luce, R. Bush, and E. Galanter (eds.), *Handbook of Mathematical Psychology*, Vol. 2, New York, John Wiley.
- Nijholt, A. (1980) *Context-free Grammars: Covers, Normal Forms, and Parsing*. New York, Springer-Verlag.
- Oehrle, R. (1975) *The Grammatical Status of the English Dative Alternation*, PhD dissertation, MIT Department of Linguistics.
- Pippenger, N., and Fischer, M. (1979) Relations among complexity measures. *J. Assoc. Comput. Mach.*, 26, 361–381.
- Posner, M., and Mitchell, R. (1967) Chronometric analysis of classification. *Psychol. Rev.*, 74, 392–409.
- Pratt, V., and Stockmeyer, L. (1976) The power of vector machines. *J. Comput. System Sci.*, 12(2), 198–221.
- Pylyshyn, Z. (1979) Validating computational models: A critique of Anderson's indeterminacy of representation claim. *Psychol. Rev.*, 86, 383–394.
- Reynolds J. (1968) *Grammatical Covering*, Chicago, Ill., Argonne National Laboratory TM 96.
- Richards, W. (1978) Quantifying sensory channels: generalizing colorimetry to orientation and texture, touch, and tones. *Sensory Processes*, 3, 207–229.
- Richter, J., and Ullman, S. (1980) A model for the spatio-temporal organization of X and Y-type

- ganglion cells in the primate retina. *MIT Artificial Intelligence Laboratory Memo No. 573*. Cambridge, MA.
- Ross, J. (1967) *Constraints on Variables in Syntax*. PhD dissertation, MIT Department of Linguistics.
- Rounds, W. (1975) A grammatical characterization of the exponential time languages. Pp. 135–143 in *Proc. 14th An. Symp. Switching and Automata Theory*.
- Ruzzo, W. (1980) On uniform circuit complexity. Pp. 312–318 in *Proc. 20th An. Conf. Foundations of Computer Sci.*
- Savin, H. B., and Perchonock, E. (1965) Grammatical structure and the immediate recall of English sentences. *J. verb. Learn. verb. Behav.*, 4, 348–353.
- Silver, W. (1981) On the representation of angular velocity and its effect on the efficiency of manipulator dynamics computation. *MIT Artificial Intelligence Laboratory Memo No. 622*. Cambridge, MA.
- Slobin, D. (1966) Grammatical transformations and sentence comprehension in childhood and adulthood. *J. verb. Learn. verb. Behav.*, 5, 219–227.
- Townsend, J. T. (1974) Issues and models concerning the processing of a finite number of inputs. *Human Information Processing: Tutorials in Performance and Cognition*. Hillsdale, NJ, Lawrence Erlbaum.
- Tyler, L. (1980) Serial and interactive parallel theories of syntactic processing, *MIT Center for Cognitive Science Occasional Paper #8*. Cambridge, MA.
- Tyler, L., and Marslen-Wilson, W. [1977] The On-line Effects of Semantic Context on Syntactic Processing, *J. verb. Learn. verb. Behav.*, 16, 683–692.
- Ullman, S. (1979) *The Interpretation of Visual Motion*. Cambridge, MA, MIT Press.
- Valian, V. (1979) The wherefores and therefores of the competence/performance distinction. In Cooper and Walker (*op. cit.*) pp. 1–26.
- Walker, E., Gough, P., and Wall, R. (1968) Grammatical relations and the search of sentences in immediate memory. *Proc. Mid-western Psycholog. Assoc.*
- Watt, W. C. (1970) On two hypotheses concerning psycholinguistics. Pp. 137–220 in J. Hayes (ed.), *Cognition and the Development of Language*. New York, John Wiley.
- Williams, E. (1975) Small clauses in English. In J. Kimball (ed.), *Syntax and Semantics*. New York, Seminar Press.

#### Reference Notes

1. Berwick, R. (1981b) *The Formal Language Theory of Lexical Functional Grammar*. LSA-ACL joint session, December, 1981.
2. Berwick, R. (1982) *Locality Principles and the Acquisition of Syntactic Knowledge*. PhD dissertation, MIT Department of Computer Science and Electrical Engineering.
3. Bresnan, J. (1972) *Theory of Complementation in English Syntax*. PhD dissertation, MIT Department of Linguistics.
4. Forster, K. (1976) *The Autonomy of Syntactic Processing*. Paper presented at the Convocation on Communication, MIT.
5. McMahon L. (1963) *Grammatical Analysis as Part of Understanding a Sentence*. PhD dissertation, Harvard University.
6. Weinberg, A. (1979) *Grammatical Theory and Parsing Theory*. Unpublished paper delivered at the 1979 GLOW Conference, Nijmegen.

#### Résumé

Dans cet article on traite de la question de savoir si et comment on peut dire que les grammaires proposées par les linguistes peuvent être actualisées en modèles adéquats de traitement de phrases. On



étudie d'abord les postulats qui guident les expériences s'appuyant sur la théorie dite de complexité dérivationnelle (DTC). Ces expériences ont été censées montrer que la théorie de la Grammaire Transformationnelle (TG) connue comme Théorie Standard n'était que partiellement adéquate pour rendre compte de l'analyse humaine. En particulier, on a pensé (voir Fodor, Bever et Garrett, 1974) que les expériences DTC démontraient que, tandis que l'analyseur utilisait les descriptions structurelles implicites dans les dérivations transformationnelles, les computations qu'il faisait ressemblaient peu aux transformations proposées par une TG. Les principales propositions sous-tendant la DTC étaient que 1) le modèle de calcul (ou analyseur) effectue les opérations de façon linéaire et sérielle et que 2) il incorpore une grammaire plus ou moins représentable sous une forme semblable à une grammaire de compétence.

Si l'on fait l'hypothèse d'une sérialité, stricte, il paraît plus facile d'inclure dans le modèle d'analyse une grammaire lexicale étendue telle que celle proposée par Bresnan (1978) comme opposée à une TG. Cette conjoncture joue un rôle important dans la critique que fait Bresnan à la TG en tant que partie pertinente d'une théorie d'utilisation du langage. Fodor, Bever et Garrett (1974) ainsi que Bresnan (1978) cherchent à rendre les règles grammaticales compatibles avec les données psycholinguistiques et avec la proposition (1). Ils proposent des modèles qui limitent la part de traitement actif réalisé en temps réel. Pour cela ils éliminent le composant transformationnel. Nous montrons que le calcul en temps réel n'est pas nécessairement associé à une complexité supplémentaire de temps de réaction. C'est à dire que nous montrons qu'un analyseur qui relie la SP à la SS par des règles de transformation (ou plus précisément par des règles d'analyse de forme très proche des règles d'un modèle transformationnel) peut s'accorder avec les données de la psycholinguistique si l'on fait simplement varier le postulat (1). Plus précisément nous montrons qu'en enchassant TG dans une architecture de calcul parallèle (qui peut être justifiée comme raisonnable pour l'usage du langage) on peut saisir les différences de complexité dans le calcul des phrases qu'avaient relevées les expérimentateurs DTC.

La proposition (2) permet aussi d'évaluer les grammaires candidates pour une théorie de l'utilisation du langage. On montre d'abord que Bresnan (1978) peut affaiblir cette proposition pour rendre l'Extended Lexical Grammar compatible avec les résultats de la psycholinguistique. Ensuite on analyse la position de Tyler et Marslen-Wilson (1977-1980) selon laquelle leurs expériences montrent qu'on ne peut instancier une TG dans un modèle d'analyse sans changer la proposition (2). Ceci est lié au fait qu'ils insistent sur le fait que leurs expériences supportent un "mode interactif" d'analyse dont ils pensent qu'il est compatible avec la Thèse de l'Autonomie de la Syntaxe. On montre que la Thèse de l'Autonomie est sans relation avec ce modèle interactif. Adopter ce modèle n'empêche donc pas d'inclure directement le TG dans un analyseur.

En outre, nous montrons pourquoi en allant dans le sens de la proposition (2), une condition que nous appelons le TTH n'est pas un critère absolu pour juger de l'utilité d'une théorie grammaticale en vue de construire une théorie d'analyseur. Nous soutenons que les grammaires ne doivent pas être envisagées comme fournissant directement et de façon transparente (proposition 2 ci-dessus), un algorithme d'analyse. Cependant, nous insistons sur le fait que la théorie de la grammaire a une place centrale dans le développement d'un mode d'utilisation du langage même si le Type de Transparence est affaibli selon nos suggestions. Enfin, nous montrons que toutes ces remarques servent à l'évaluation comparative des modèles d'analyses possibles qui incorporent la grammaire transformationnelle, la grammaire lexico-fonctionnelle et les propositions de Tyler et Marslen-Wilson.