

## Chapter 3

# A Speculation about what Linguistic Structures Might Be

Avery Andrews

A feature of generative linguistics that might seem rather strange to those in other areas of cognitive science is the ubiquitous ‘structures’; syntactic, phonological, semantic, etc., ascribed to utterances.<sup>1</sup> As far as I know, these tend to lack clear counterparts in for example many areas of psychology (but not all, such as the analysis of the visual system by Marr (1982)), while for linguists they appear to be part of the furniture of their childhood home, and there is relatively little discussion what they are really about.

In this chapter, I will propose a speculation to the effect that they can be regarded as a combination of two ideas. One, relatively old, is that they are abstract objects from which various properties of sentences can be easily computed, including, but not limited to, overt form, possible interpretation, and additional properties such as sociolinguistic register (e.g. Arka 2005) or genre. The other, relatively new, is that they are equivalence classes of (often partial) proofs or calculations that a sentence has those properties. An important aspect of this partiality is that they can represent a shared portion of the calculation of different properties, such as literal meaning and register or genre, which would use the same path for a while, and then diverge.

The first view is based on musings that I had as a graduate student and early career syntax teacher, whereas the second is from a more recent interest in proof theory, especially as used in glue semantics.

---

<sup>1</sup>I would like to thank the two reviewers for very extensive and helpful comments.

## 3.1 Early Generative Practice

### 3.1.1 Phrase Structure

A simple example of both views at the same time is provided by phrase-structure (PS) trees, which emerged spontaneously from Chomsky's conception of a P-marker, perhaps because the closest thing to an intelligible representation of a P-marker is indeed a tree; the difference is discussed below. Some of the important characteristics of a PS tree are:

- (1) a. It determines a string via the 'yield' function that produces a list of its leaves, in order.
- b. A property of 'grammaticality' can be calculated easily by seeing what sequence of daughters each node has (not adequate to fully model acceptability in any given human language, but often a pretty good start).
- c. A tangible relationship to many aspects of semantic interpretation, for example via lexical entries for verbs assigning semantic roles to NPs in their vicinity, using the techniques of 'selection restrictions' in Chomsky (1965), and the later versions of Katz and Fodor's approach to semantics, such as Katz (1972) (however, for a full story, a more abstract level of representation is also needed).<sup>2</sup>
- d. Given a string and a PS grammar, the set of structures that have that string as yield can be enumerated, and if the grammar obeys some reasonable conditions, the list of structures will be finite.
- e. Further properties not conventionally identified as grammaticality can be determined, such as excessive left or center embedding (Karlson 2009, 2010) and others.

Trees emerge intuitively from depictions of the process whereby a phrase-structure grammar (PSG) defines a set of strings: starting with an 'initial symbol', typically 'S' for 'sentence', one would write an instance of S at the top of the diagram, and then one of sequences of phrase types specified by the PS rules as a possible expansion/sequence of daughters of S, and then similarly for the daughters, each connected to the mother by a line.

---

<sup>2</sup>What is exactly the status of 'selection restriction violations' has been controversial for a long time; in some versions of early generative grammar they were considered 'ungrammatical', although not necessarily to a high degree (Chomsky, 1964), but for some older arguments that they should be considered as grammatical, see Shanon (1976), and a more recent and extensive development pointed out by a reviewer, Magidor (2013).

I am not aware of any thoroughly developed account of why it was seen as such a good idea to draw pictures, but the practice was and is compelling, and there were various proposals for drawing sentence diagrams in circulation in the 1950s and before, perhaps going back to the Reed-Kellogg diagrams that used to be taught in secondary school. Interestingly, Chomsky himself did not formally advocate the use of trees, but rather a different idea, ‘P-markers’.

A P-marker was supposed to be a set of strings, representing all the ‘equivalent’ ways of deriving a sentence. Very few people paid much attention to this concept (even Chomsky himself frequently drew trees, with no explanation of the difference between these and P-markers, even calling a tree a P-marker in Chomsky (1965, 65)), with the interesting exception of Lasnik & Kupin (1977), but I recall (from a class lecture by Chomsky) that one of its features was supposed to be that it represented ‘is a’ facts about subsequences of utterances, without dragging in certain additional putatively irrelevant additional distinctions. For example, in a sentence such as *that Maurice loved bagels surprised Louise*, on the assumption that the subject clause was both an NP and an S, a standard tree would represent the S as under the NP, whereas a P-marker would not, merely recording the information that *that Maurice loved bagels* ‘is a’ NP and ‘is a’ S. So the tree overspecified certain aspects of the intuitions associated with the sentence, while the P-marker did not.

### 3.1.2 Transformational Grammar

In Transformational Grammar, phrase structure was of course only a prelude to and part of the full theory, which existed in a series of forms, the first two being the original ‘Syntactic Structures’ (SS, Chomsky 1957) model, in which multiple trees (of ‘kernel sentences’) were developed and combined into a surface/overt structure by transformations, and the later ‘Aspects of the Theory of Syntax’/Katz-Postal model (ATS/KP, Chomsky 1965, Katz & Postal 1964), in which a single deep structure tree determined the meaning, and was processed by the transformations into a surface structure. SS derivations met (or didn’t meet) characteristics (a-d) of (1) as follows:

- (2) a. The string is the PS yield of the final tree in the derivation.
- b. A structure is grammatical if the initial trees satisfy the PS rules, and the final tree is produced by applying the transformations to them obeying the ordering constraints.
- c. There wasn’t any explicit account of meaning.
- d. If an SS theory was fully formalized in some way, it would clearly be possible to recursively enumerate the structures that it assigned to

a given string, but practical parsing does not appear to have been a possibility.

With ATS/KP, things are similar but not quite the same:

- (3) a. The string is the PS yield of the final tree in the derivation.
- b. A structure is grammatical if the initial tree satisfies the PS rules, and the final tree is produced by applying the transformations to it obeying the ordering constraints, and any further constraints that might be imposed (such as ‘#’ erasure for relative clauses).
- c. Semantic interpretation was to be read off the initial (‘deep structure’) tree. (This was basically Katz and Postal, although Chomsky accepted it for a while, but then concluded (Chomsky, 1970) that semantic interpretation needed to look at both deep and surface structure.)
- d. Considerable attention was devoted to assuring that the list of structures assigned to a string would be finite (‘recoverability of deletion’), and there were substantial efforts to produce parsers with a ‘covering grammar’ hand-produced by graduate students (e.g. Plath 1973). Later developments included the ‘Structure Preserving Constraint’ of Emonds (1970,1976), which improved the prospects for parsing.

The most important aspect of the transition from SS to AST/KP was the addition of the goal of providing an account of semantic interpretation in generative grammar. Both versions supported a version of (1e) in the form of the ‘Derivational Theory of Complexity’, the idea that other things being equal, a sentence with a longer derivation should show signs of being more difficult to process than a shorter one. Actual arguments about this appear to have run into numerous difficulties, which I won’t try to review here.

There was as far as I know never much discussion of the role of sentence structures in early TG, but it is clear that further kinds of properties such as those mentioned in (1e) can also be read off a transformational derivation, which we can think of as aspects of ‘meaning in a wider sense’, not worrying about how to distinguish semantics from pragmatics and further niceties. Discourse properties such as topic and focus, etc, have been fully mainstream for quite some time now, often encoded in discourse Projections such as TopP and FocP, and while there has not been so much discussion of register and genre, it is clear that these and similar properties can be signalled by syntactic structure as well as by lexical and morphological choice. For example, the archaic/elevated/poetic register of constructions such as:

- (4) a. Ask not what your country can do for you.

b. How green the fields, how blue the sky.

In (a), the archaic feature is putting the negative after the main verb, without supporting *do*; in (b), putting the predicate adjective phrase in initial position, followed by the subject, with no copula. Neither of these constructions can be used today in serious new compositions.<sup>3</sup> There appears to be no limit in principle to the types of properties that might be read off a syntactic structure.

### 3.1.3 Later Developments

Transformational Grammar in its earlier forms could be regarded as being somewhat schizophrenic in its treatment of structures, with the rather compact tree representation of the initial PS derivation and derived structures contrasting with the more diffuse and highly redundant transformational derivation. It is therefore perhaps not a surprise that later versions of TG tended to involve ideas such as the Structure Preserving Constraint and conventions for connecting deletion or movement sites to their controllers with lines, leading to widespread use of more compact representations, but without the emergence of a clear theoretical consensus that the results were truly correct (Hunter, 2019).

The ‘Alternative Generative Theories’ on the other hand, such as LFG, HPSG, and certain forms of Cognitive or Construction Grammar,<sup>4</sup> tended to prefer more compact formats with greater similarity to trees than to derivations. It is perhaps interesting to note that the extensive use of the  $\uparrow = \downarrow$  causes LFG to have considerably more compact linguistic structures than HPSG or various forms of construction grammar, where the feature structure of a mother node is complex, and computed from those of its daughters by various principles. But this is a matter of degree, since certain developments in LFG such as restriction and distributive features reduce this distinction between the approaches.

Another characteristic of the later work was emphasis on showing that the problem of associating a sentence with its structures including meaning was at least recursive and ideally tractable,<sup>5</sup>

---

<sup>3</sup>According to my recollections, there was no widespread criticism of Kennedy for using this construction in 1960, but I suspect that if a political figure were to deploy it today, they would be subject to extreme ridicule. So this is a change in status of a construction in the last 60 years from ‘archaic, but still usable for serious purposes’, to ‘only usable for certain minor forms of entertainment, such as satire, Tolkien imitations, and similar’.

<sup>4</sup>Such as Kay & Fillmore (1999) for something not fully formalized, but comparable in degree of formalization to most work in Transformational Grammar and its descendants, or ‘Sign-Based Construction Grammar’ (Boas & Sag, 2012) for something much more formally developed.

<sup>5</sup>Unfortunately, current LFG is not tractable, although some possible changes to make it so are discussed in Kaplan & Wedekind (2020). It is also possible that tractability is a bit overrated; some of the calculations used to make ‘corridor shooters’ such as the computer games *Doom* and *Quake* playable on the slow computers of the 1990s were intractable, but that did not stop

A feature of the compact structures is that, typically, multiple constraints can be checked for them asynchronously, which is not the case for derivations in general. This leads to the distinction made by Pullum & Scholz (2001) between ‘Model Theoretic’ (MTS) and ‘Generative-Enumerative’ (GES) approaches to syntax. In MTS, represented in a relatively pure form by HPSG, a collection of structures is defined by producing a set of constraints which a well-formed structure needs to satisfy, whereas in GES, exemplified in a relatively pure form by many versions of Minimalism, one simply gives a procedure for producing the structure. I perhaps disagree a bit with Pullum and Scholz in seeing this more as a gradient than a division.

For example, PS Grammar can be presented either way, but I think that the GES view can be made intelligible to a top-quality undergraduate student by starting to draw a diagram and talking about it for about a minute and a half, while good undergraduates might require the picture to be finished, along with perhaps another two minutes or so of narrative. An MTS presentation on the other hand would be considerably harder for even the best students, and very likely out of reach for many of the others. This indicates that although PS Grammar can be developed in the style of MTS, a GES presentation has significant virtues.

TG itself seemed to take a somewhat hybridized view. For example Perlmutter (1971) had conventional transformational derivations supplemented with MTS-like constraints at both the deep and surface levels, whereas Lakoff’s ‘Global Rules’ (1970) invited thinking of a transformational derivation as an object in an MTS-like style. This will relate to an issue we discuss later, the rather ‘aspirational’ nature of linguistic structures as representations of equivalence classes of computations a.k.a. proofs.

## 3.2 Proof Equivalence

As I have been arguing, the prevailing generative view has been that a structure is a convenient object from which various properties can be read off. A somewhat different approach is implicit in the trees of PSG, but only developed explicitly in approaches to syntax and semantics derived from Proof Theory, such as the varieties of Categorical Grammar, and the Glue Semantics of LFG, introduced in Dalrymple et al. (1993), with most of the work from the 1990s collected in Dalrymple (1999), and the current ‘new glue’ system codified in Dalrymple (2001). The basic idea of proof equivalence is that it is sometimes desirable to regard two superficially different proofs of the same thing from the same premises as being the same proof.

---

thousands of maps for these games from being made and played; mappers just used techniques such as ‘vis-blockers’ to avoid the difficult cases.

The simplest concrete example for linguists, essentially noted by Chomsky, is phrase-structure derivations. Suppose we have the grammar of (5), and the two derivations of (6) (lexical development omitted):

- (5) a. S → NP VP  
 b. NP → Det N  
 c. VP → V VP

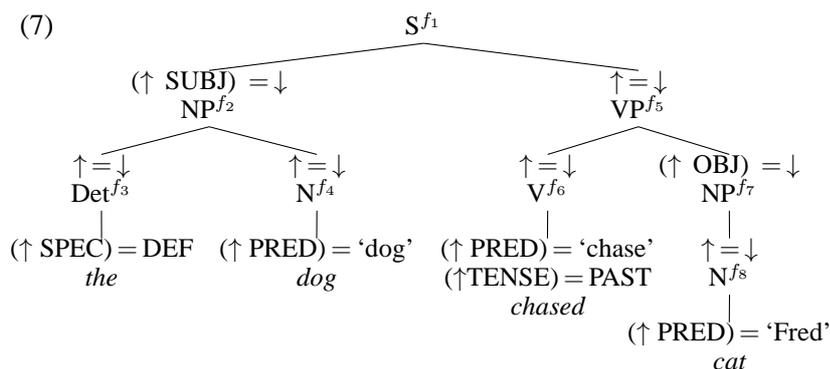
- (6) a. S  
 NP VP  
 Det N VP  
 Det N V NP  
 Det N V Det N
- b. S  
 NP VP  
 NP V NP  
 NP V Det N  
 Det N V Det N

This difference in derivation does not appear to represent any interesting property of the sentence, and the notion of tree-building as discussed previously provides a notion of equivalence of derivation that renders them both equivalent, providing further benefits such as for example a connection to semantics, as first discussed systematically in the Aspects/Katz-Postal framework. If we think of a PS derivation as a proof that a string is generated by a grammar, the tree is then a representation of an equivalence class of such proofs.

Note however that Chomsky's original notion of P-markers would probably not strike logicians as an appropriate notion of proof-equivalence, since it doesn't distinguish between the use of a premise  $A \rightarrow B$  from that of  $B \rightarrow A$ . Linguists appear to have chosen intuitively the version of proof-equivalence that requires that equivalent proofs use exactly the same premises, and that the structures represent equivalent proofs/computations rather than display of equivalent linguistic data. Classic Transformational derivations also had some limited possibilities for proof/derivation equivalence, such as the order in which subordinate clauses were processed before being combined into a main clause (either by generalized transformations in SS, or cyclical rule application in AST/KF), but not much was made of this.

A later example is LFG's f-structures and solution algorithm, in Kaplan & Bresnan (1982). An f-structure can be regarded as a representation of a set of proofs that the equational annotations to an LFG annotated PS structure (c-structure) are consistent. The proofs in the set differ in the order in which the

equations are processed, a difference which we wish to disregard. To view the f-structure as a proof, however, we need to associated it with the annotations associated with the c-structure by the annotated PS rules, using the labels introduced in instantiation to provide the association. For example in (7) below, the c-structure nodes are given labels which are referred to by the arrows, resulting in the f-description of (8) whose consistency is proved by (9):



(8)  $(f_1 \text{ SUBJ}) = f_2$   
 $f_2 = f_3$   
 $(f_3 \text{ SPEC}) = \text{DEF}$   
 $\vdots$

(9) 
$$\left[ \begin{array}{l} \text{SUBJ} \quad f_2, f_3, f_4 \left[ \begin{array}{l} \text{SPEC} \quad \text{DEF} \\ \text{PRED} \quad \text{'dog'} \end{array} \right] \\ f_1 \text{ TENSE} \quad \text{PAST} \\ f_5 \text{ PRED} \quad \text{'chase'} \\ f_6 \text{ OBJ} \quad f_7, f_8, \left[ \text{PRED} \quad \text{'Fred'} \right] \end{array} \right]$$

See Kaplan & Bresnan (1982) for the details of the solution algorithm, but the basic idea is to build the f-structure by consulting the equations in sequence, halting with failure (proof of inconsistency) if you need to specify different attributes for some value, for example conflicting number on the subject in an example such as *\*the cats likes the dog*. Similarly to trees, it is also the case that we can use the f-structure with labels to check that it satisfies all of the equations in the f-description, which are thereby shown to be consistent. This of course is very much like a model-theoretic proof of consistency, putting this aspect of LFG firmly within the Pullum-Scholtz concept of MTS.

The proofs that we wish to regard as equivalent are the ones that differ only in the order of processing of the annotations. Kaplan (1987)<sup>6</sup> proposes that

<sup>6</sup>p. 345 in the reprinted version in Dalrymple et al. (1995)

this is desirable because it supports parsing out from ‘islands of certainty’ in the speech stream, rather than requiring strict left-to-right processing (which, however, has its strong advocates).

There are two further aspects of f-structures that need comment. One is that the f-structure is supposed to be the minimal<sup>7</sup> solution to the equations: there will in general be more solutions containing additional material. For example if we added a CASE ACC attribute to (8), we would still have a proof that (7) were consistent. But it would clearly be invalid to toss extra structure into the developing f-structure in this manner, due to the likelihood of falsely detecting inconsistency. Another is the requirement that the f-structure be unique as well as minimal. This played a somewhat technical role in the 1982 version of the theory, and it is not entirely clear that we need it today, but the basic idea was to rule out structures produced the ‘PCASE’ analysis of prepositional arguments presented in Kaplan & Bresnan (1982, 196-201), which would have an unspecified variable as an attribute. This would be a final check on a proof that would need to give the same result for all members of the equivalence class.

A further example is provided by glue semantics. This is based rather heavily on the idea of proof-normalization. Glue proofs can in general be presented in multiple ways, involving different order of application and ‘detours’, such as chains of application of implication introduction and elimination in natural deduction. Without proof-normalization steps such as  $\beta$ -contraction and, if desired,  $\eta$ -contraction, there won’t be a finite number of semantic structures associated with the string. Proof-nets provide a diagrammatic portrayal of these equivalence classes, although they don’t appear to have been especially popular.<sup>8</sup> Tree-style natural deductions similarly abstract away various irrelevant differences in application of proof-steps, but contain considerable internal redundancy. Redundancy versus easy checkability by humans may be a tradeoff that is difficult to remove.

We have seen then that in many cases, a concept of ‘equivalent proofs’ can be seen as playing an important role, and is representable as some kind of structure. In the final section I will attempt to combine this with the idea of the previous section that a linguistic structure is something that can be calculated, and from which various linguistic properties can be read off, to produce the idea that a linguistic structure (of sentence or utterance) represents an equivalence class of (possibly partial) computations of properties of sentences or utterances.

---

<sup>7</sup>The notion of ‘minimality’ can be understood as that of being the minimal element in an ordering of f-structures by subsumption; see Johnson (1988) for discussion of many of the relevant issues.

<sup>8</sup>And I am not aware of any proposals to extend them to monads, proposed as an addition to glue by Asudeh & Giorgolo (2016) and Asudeh & Giorgolo (2020).

### 3.3 Structures as Equivalence Classes of Computations

An important important development in recent times has been the blurring, essentially erasure, of the difference between certain kinds of proofs, and computation. The Curry-Howard Isomorphism for example can be seen as the discovery that the computational technique of the lambda calculus has the same structure as proof simplification in certain logical systems. Generally, various techniques for proving that an object has a certain property or value of a property tend to be nondistinct from techniques for calculating the value of that property, or whether the object in question has it (especially since the appearance of practical logic programming systems in the 1980s).

We can therefore see a linguistic structure as an attempt to represent an equivalence class of computations of linguistic properties, ideally extending to both parsing and production. However, there is a major difference between the sentence (or utterance) structures in linguistics and the classes of equivalent proofs in logic, which is that the latter are ‘actual’, in the sense that we know exactly what kinds of proofs we are trying to put into the same class (in any given instance, however the question of why those is perhaps not always so clear), while the former are ‘aspirational’: we do not know what the computations are actually like, but are only pursuing some ideas about how they might be organized, and that under some assumptions that might be questioned, such as that the essential organization (but not of course the detailed flow) is the same for understanding and production. I will try to illustrate this theme by discussing morphology and the famous ‘Person-Case Constraint’ in LFG.

A typical example is provided by preverbal ‘clitic’ object pronouns in Modern Greek, as first discussed in generative grammar by Bonet (1991, 181). In this language, if there is an ‘indirect object’ clitic ( $OBJ_{\theta}$  according to Kordoni (2004), although the literature on Greek double object constructions is not extensive in LFG, and I am not aware of any worked out applications of Lexical Mapping Theory to this language), then any direct object clitic cannot be first or second person object, but only third. If the preverbal clitics are analysed as prefixes to the verb, as usually proposed (Condoravdi & Kiparsky, 2001), we can express the Greek version of the PCC with the following constraint on the f-structure that is found by solving the annotations of the verb:

$$(10) * \left[ \begin{array}{l} OBJ \quad [PERS \quad I/II] \\ OBJ_{\theta} \end{array} \right]$$

This delivers the result that (a) is good, because the  $OBJ$  *ton* is not first or second person, but (b) is bad, because the object *me* is first person, while (c) is good again, because the direct object is not doubled as a clitic:

- (11) a. tha su ton stílune  
 FUT you(DAT) him(ACC) send.3pl  
 They will send him to you.
- b. \*tha tu me stílune  
 FUT him(DAT) me(ACC) send  
 They will send me to him.
- c. tha tu stílune eséna  
 FUT him(DAT) send you(ACC)  
 They will send you to him.

Note that in Greek, clitic doubling of accusative pronouns is optional, dative ones obligatory, giving us a simple ‘repair strategy’ for (\*b), which is to remove the clitic ‘me’ and add the free object pronoun ‘eména’, parallel to ‘eséna’ in (c).

This is the so-called ‘strong’ version of the PCC; many other languages such as Italian, Catalan and Occitan show a weaker version, in which first or second person direct objects are acceptable as long as the indirect object is not third person (Anagnostopoulou, 2017, 3-4). This can be accommodated with a slightly elaborated version of (10):

- (12) \*[OBJ [PERS I|II], OBJ<sub>θ</sub> [PERS III]]

There are also further variations; the ‘Me-First PCC’ (Rumanian) blocks 2/3 person IO when DO is 1st person:

- (13) \*[OBJ [PERS I], OBJ<sub>θ</sub> [PERS II|III]]

Especially interesting from the point of view of how these constraints should really be formulated, is the ‘Ultrastrong’ version, first noted in Fassi Fehri (1988) for Arabic, but also found in some varieties of Catalan. Descriptively, here, the OBJ<sub>θ</sub> (IO) must outrank the OBJ on the person hierarchy I>II>III. Therefore (13) is ambiguous for weak speakers, but has only the first reading for Ultrastrong ones:

- (14) Te’m van recomenar per la feina  
 II I recommended for the job  
 They recommended you to me for the job.  
 They recommended me to you for the job.

I won’t attempt to state this version of the constraint formally, since that would lead on to numerous issues concerning hierarchies and the internal structure of feature-values, but it shares with the other variants the requirement (for an LFG treatment) to assemble the f-structure information associated with a single

word, or word-like unit, and apply constraints to this, before assembling all of the results.

The qualification ‘or word-like unit’ is needed because defining the word is one of the classic unsolved problems of linguistics; the word-like concept relevant here is not necessarily the same as the one involved in morphological irregularities such as suppletion and less drastic forms of allomorphy involved with stems and affixes. I will call this unit the ‘syntactic word’, since it seems to impose limitations on what kinds of f-structure features can be expressed in whatever kind of word-like unit it is ultimately judged to be.<sup>9</sup> The syntactic word might also be involved in phenomena such as morphological blocking as discussed by Andrews (1990).

On this basis, it would be reasonable to somewhat elaborate the notion of ‘sentence structure’ in LFG to include f-structures for each individual syntactic word, as proposed and called the ‘ $\lambda$ -projection’ by Kuhn (2003), as part of the infrastructure for Optimality-Theoretic LFG. This idea can be seen as implicit in the standard presentation of an annotated f-structure with the lexical annotations written in association with the individual lexical items, but if we are going to use it for any purpose, it is better to make it explicit, since it constitutes a structure level that the PCC can apply to.

This idea is highly compatible with recent work in LFG morphology, as reviewed and integrated by Dalrymple et al. (1995), and, more recently, Dalrymple (2015). In this work it is proposed that there is a separate morphological component which produces morphological forms associated with f-structure information. In this context, the PCC provides evidence that we solve the f-structural information provided by each word individually, before combining the results to produce that of the sentence. On the other hand, it provides no evidence concerning existence of ‘m-features’ associated with the morphology, but converted into f-structure features for the production of the f-structure.

### 3.4 Conclusion

Regarding linguistic structures as aspirational equivalence classes of proofs or computations provides a notion of what they are about that relates them to computation but does not make excessively strong claims. It is also consistent with taking a rather provisional attitude towards their details. In glue semantics, for example, proof nets and (tree-style) natural deductions can be regarded as equating the same proofs, but presenting them in different formats. We can also regard the distinction between Generative-Enumerative and Model Theoretic approaches to syntactic structures as matters of convenience rather than

---

<sup>9</sup>See Preminger (2019) for discussion of why the PCC can’t be strictly morphological, but does depend on exactly what features a form is expressing.

fundamental importance.

This conception can perhaps also be regarded as a progression on the concept of ‘level 1.5’ presented in Peacocke (1986, 1989) and Davies (1987) as an elaboration of Marr’s (1982) levels 1, 2 and 3 of explanation in cognitive science. Level 1 is supposed to be an extensional specification of what some mental computations do, 2 a specification of the algorithms which they use to do it, and 3 an account of how these algorithms are neurologically implemented. Level 1.5 is a suggested interpolation intended to represent the knowledge used by the algorithm, intermediate between what the algorithm does, from a strictly extensional view, and what it actually is, as a procedure. For example, writing multiple versions of the phrase structure rules for NP in a language with NP-internal agreement would almost certainly be wrong at level 1.5, since it is very unlikely that the apparent PS rules for NPs with different agreement feature specifications would retain the same form for long if there were multiple specifications in use rather than just one.

With the structures however, we try to go a bit further in characterizing the organization of the processes involved, without fully specifying an algorithm. In particular, an argument for a linguistic level can be seen as an argument for a possibly shared pathway for computation of different properties, such as literal meaning and register/genre in the archaic examples of (4), and the nature of significant stages along that pathway. Perhaps this could be called ‘level 1.6’, a further step from 1.5 along the way to 2. Observe in particular that postulating  $\lambda$ -structure does not add any specific information to an LFG grammar (which has the same information in the same format with or without  $\lambda$ -structure), nor does it specify any specific algorithm for using that information, but it does impose a constraint on the organization and functioning of such algorithms.

## Bibliography

- Anagnostopoulou, Elena. 2017. The person-case constraint. In Martin Everaert & Henk van Reimsdijk (eds.), *The wiley-blackwell companion to syntax*, Wiley Blackwell 2nd edn.
- Andrews, Avery D. 1990. Unification and morphological blocking. *Natural Language and Linguistic Theory* 8. 507–557.
- Arka, I Wayan. 2005. Speech levels, social predicates and pragmatic structure in Balinese: A lexical approach. *Pragmatics* 15. 169–203.
- Asudeh, Ash & Gianluca Giorgolo. 2016. Perspectives. *Semantics & Pragmatics* 9.
- Asudeh, Ash & Gianluca Giorgolo. 2020. *Enriched meanings: Natural language semantics with category theory*. Oxford University Press.

- Boas, Hans C. & Ivan A. Sag. 2012. *Sign-based construction grammar*. CSLI Publications.
- Bonet, Eulalia. 1991. *Morphology after syntax*: Massachusetts Institute of Technology dissertation.
- Chomsky, Avram Noam. 1964. Degrees of grammaticalness. In Jerry A. Fodor & Jerrold J. Katz (eds.), *The structure of language: Readings in the philosophy of language*, 384–389. Prentice-Hall.
- Chomsky, Noam. 1957. *Syntactic structures*. Mouton.
- Chomsky, Noam. 1965. *Aspects of the theory of syntax*. MIT Press.
- Chomsky, Noam. 1970. Deep structure, surface structure and semantic interpretation. In Roman Jakobson & Shigeo Kawamoto (eds.), *Studies in general and Oriental linguistics presented to Shiro Hattori on the occasion of his sixtieth birthday*, 52–91. TEC Company, Ltd. Reprinted by IULC and Mouton as Chomsky (1972) *Studies in Semantics in Generative Grammar*.
- Condoravdi, Cleo & Paul Kiparsky. 2001. Clitics and clause structure. *Journal of Greek Linguistics* 2. 1–39.
- Dalrymple, Mary (ed.). 1999. *Semantics and syntax in Lexical Functional Grammar: The resource logic approach*. MIT Press.
- Dalrymple, Mary. 2001. *Lexical functional grammar*. Academic Press.
- Dalrymple, Mary. 2015. Morphology in the LFG architecture. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of LFG15*, 64–83. CSLI Publications.
- Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell & Annie Zaenen (eds.). 1995. *Formal issues in Lexical-Functional Grammar*. CSLI Publications.
- Dalrymple, Mary, John Lamping & Vijay Saraswat. 1993. LFG semantics via constraints. In Michael Moortgat, Steven Krauwer & Louis des Tombe (eds.), *Proceedings of the sixth conference of the european ACL*, 97–105.
- Davies, Martin. 1987. Tacit knowledge and semantic theory: Can a five percent difference matter? *Mind* 96. 441–462.
- Emonds, Joseph. 1970. *Root and structure-preserving transformations*: Massachusetts Institute of Technology dissertation. Distributed by the Indiana University Linguistics Club.
- Emonds, Joseph. 1976. *A transformational approach to English syntax*. Academic Press.
- Fassi Fehri, Abdelkader. 1988. Agreement in Arabic, binding and coherence. In Michael Barlow & Charles Ferguson (eds.), *Agreement in natural language*, 107–158. CSLI Publications.
- Hunter, Tim. 2019. What sort of cognitive hypothesis is a derivational theory of grammar? *Catalan Journal of Linguistics* 89–138.
- Johnson, Mark. 1988. *Attribute-value logic and the theory of grammar*. CSLI Publications.

- Kaplan, Ronald M. 1987. Three seductions of computational psycholinguistics. In Peter Whitelock, Mary McGee Wood, Harold L. Somers, Rod Johnson & Paul Bennet (eds.), *Linguistics and computer applications*, 149–188. Academic Press.
- Kaplan, Ronald M. & Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan (ed.), *The mental representation of grammatical relations*, 173–281. MIT Press.
- Kaplan, Ronald M. & Jürgen Wedekind. 2020. Zipper-driven parsing for LFG grammars. Talk delivered at the 2020 LFG Conference.
- Karlsson, Fred. 2009. Origin and maintenance of embedding clausal complexity. In Geoffrey Sampson, David Gil & Peter Trudgill (eds.), *Language complexity as an evolving variable*, 192–202. Oxford University Press.
- Karlsson, Fred. 2010. Syntactic recursion and iteration. In Harry van der Hulst (ed.), *Recursion in human language*, 43–68. Mouton de Gruyter.
- Katz, Jerrold J. 1972. *Semantic theory*. Harper and Row.
- Katz, Jerrold J. & Paul M. Postal. 1964. *An integrated theory of linguistic descriptions*. MIT Press.
- Kay, Paul & Charles J. Fillmore. 1999. Grammatical constructions and linguistic generalizations: The *what's x doing y?* construction. *Language* 75. 1–33.
- Kordoni, Valia. 2004. Modern Greek ditransitives in LMT. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of LFG04*, 307–312. CSLI Publications.
- Kuhn, Jonas. 2003. *Optimality-theoretic syntax: A declarative approach*. CSLI Publications.
- Lakoff, George. 1970. Global rules. *Language* 46. 627–639.
- Lasnik, Howard & Joseph Kupin. 1977. A restrictive theory of transformational grammar. *Theoretical Linguistics* 4. 173–196. Reprinted, with minor revisions, in Lasnik (1990) *Essays on Restrictiveness and Learnability*, Kluwer Academic Publishers.
- Magidor, Ofra. 2013. *Category mistakes*. Oxford University Press.
- Marr, David. 1982. *Vision: A computational investigation into the human representation and processing of visual information*. W.H. Freeman.
- Peacocke, C.M. 1986. Explanation in computational psychology: Language, perception and level 1.5. *Mind and Language* 1. 101–123.
- Peacocke, C.M. 1989. When is a grammar psychologically real? In A. George (ed.), *Reflections on Chomsky*, 111–130. Blackwell.
- Perlmutter, David M. 1971. *Deep and surface structure constraints in syntax*. Holt, Rinehart and Winston.
- Plath, Warren. 1973. Transformational grammar and transformational parsing in the REQUEST system. In *COLING 1973 volume 2: Computational*

*and mathematical linguistics: Proceedings of the international conference on computational linguistics, .*

Preminger, Omer. 2019. What the PCC tells us about “abstract” agreement, head movement, and locality. *Glossa* 13. 1–42.

Pullum, Geoffrey K. & Barbara C. Scholz. 2001. On the distinction between model-theoretic and generative-enumerative syntactic frameworks. In Philippe de Groote, Glyn Morrill & Christian Retoré (eds.), *Logical aspects of computational linguistics (LACL 2001)* (Lecture Notes in Computer Science 2099), 17–43. Springer.

Shanon, Benny. 1976. Semantic and syntactic features in the context of dreams. *Linguistic Inquiry* 7. 518–519.