Actual Language Use and Competence Grammars

Gregory M. Kobele

1 Introduction

In their target article, Kempson, Cann, Gregoromichelaki and Chatzikyriakidis (henceforth **KCGC**) begin with the observation that language use (production and comprehension) is incremental, and that this incrementality is particularly evident in normal dialogue situations where hearers complete speakers' sentences (and back again). They argue that data and generalizations like this militate against the

> standard methodology of isolating classes of phenomena as independent of language use and analysable through declarative propositional knowledge ('competence') invoked during online processing.

My goal in this short note is to argue that this is false. In particular, I will argue for the following:

1. that left-to-right incremental interpretation has nothing to do with the structures assigned by the grammar to strings
2. that split utterances across dialogue are easily accounted for in *any* standard grammatical framework

2 Abstractions

KCGC claim (in their section 3) that the competence-performance and the levels-of-analysis distinctions cannot reasonably be maintained in the light of split utterance data. I wish to briefly clarify what I take these distinctions to be.

2.1 Competence and Performance

The competence/performance distinction is (as I understand it) fundamentally a hypothesis about the nature of the origin of the data in some domain. In virtually every real-life situation we can imagine, what actually ends up happening is the result of the interaction of a slew of many different causal powers. For example, the particular sentence I utter in a certain situation is influenced by my mood, by my attitudes about my interlocutor, by my goals as regards the conversation, by my overall goals for the day, etc, some of which (for example mood), are themselves influenced by multiple causal powers (what I ate for breakfast, whether I slept well, whether I've been having a good week, etc). One perspective on the goals of science is that we are attempting to "carve up nature at the joints",[1] that is, trying to identify and understand the myriad causal powers engendering observables. The competence/performance distinction is simply a name for this methodology specialized to cognitive phenomena. Here, *competence* is the theory of a particular causal power implicated in some behaviour, and *performance* is the theory of how this particular causal power interacts with or is influenced by the other causal powers relevant to said behaviour. Chomsky's much maligned *ideal speaker-listener* abstracts away from

> memory limitations, distractions, shifts of attention and interest, and errors [Chomsky, 1965]

which are (or include) obvious causal powers of relevance to actual behaviour, in an attempt to identify and understand the contribution of another (the rules of grammar).

2.2 Levels

There are often multiple true descriptions of things.[2] Sometimes, these describe different aspects of the same thing, and other times the same aspect is described, but at different levels of granularity.

In the context of information processing systems (devices with input-output behaviour), it is useful to describe their behaviour at multiple levels

[1] Some might trace this back to Plato's Phaedrus 265e.

[2] Tinbergen [1963] provides an influential example of this in ethology.

[Marr, 1982].[3] At the highest (most abstract) level ('computation'), one can characterize exactly which inputs are paired with which outputs. Slightly less abstractly ('algorithm'), one can provide some details about how this pairing is achieved. Most concretely ('implementation'), one can describe the pairings of inputs and outputs in terms of the brute physical properties of the system.[4]

A computer program is the most straightforward example. We can ask what function is computed, what data structures and algorithms are used to compute the function, as well as how the program is executed on a given machine. Each of these provides important information about the program, but, for example, knowing just the data structures and algorithms used without knowing what the program was actually supposed to be doing would not be very insightful.

2.3 The nature of the disagreement

KCGC are clearly continuing to abstract away from memory limitations, distractions, shifts of attention and interest, and errors, whence they are by their practice adopting the standard competence-performance distinction, as I have described it above.

KCGC's claim must therefore be about levels, and in particular about computational versus algorithmic level descriptions of grammar (the causal power studied by linguists which is relevant to our use of language). What could it mean, to claim that certain phenomena "cannot be subsumed under [these] idealisations?" One very reasonable kind of claim is that a particular phenomenon requires for its explanation a kind of theoretical vocabulary that is available only at a lower level. One aspect of KCGC's claim seems to be just this: that these dialogue-y phenomena require for their explanation a more processing based theoretical vocabulary. I believe that this could very well be the case, and my proposal in §4 can be understood as being in agreement with this claim. However, they seem to draw from this claim the conclusion that it is never fruitful to understand grammar at a high level (i.e.

[3] Marr [1977] distinguishes situations in which there is a single causal power which dominates the phenomena from those where the phenomena emerge as the interaction of many such. A theory of the first case he calls **Type 1**, one of the second is **Type 2**. He suggests that language behaviour will require a type 2 theory, which I find plausible.

[4] Anderson [1990] adopts a similar methodology, advocating for understanding aspects of cognition as being optimized somehow for their environment.

abstracting away from the data structures and algorithms which undergird this use). I understand this conclusion as being a strong motivator for the dynamic syntax enterprise.

3 Grammar

The linguist is attempting to describe one of the (purported) causal powers underlying our ability to use (and learn) language. The standard way of approaching this task is to attempt to describe the relation between forms and their meanings that we end up computing while speaking and listening, independently of attempting to describe the on-line process by means of which this relation gets computed. (As KCGC point out, there is a bias towards treating complete sentences as somehow basic.) The form that a description of this sound-meaning relation takes is given by a(n interpreted) grammar.

In this section, I will take as a concrete example of a grammar formalism context-free grammars. While linguists [Chomsky, 1956] and computational linguists [Shieber, 1985, Huybregts, 1985] uniformly reject context-free grammars as too weak to describe the form-meaning relations in natural language, everything said here about context-free grammars extends immediately to more powerful and popular grammar formalisms like tree adjoining grammars [Joshi et al., 1975] and minimalist grammars [Stabler, 1997].

The main take-away from this section should be the following:

> compositional semantics suffices for incremental interpretation

In other words, *pace* KCGC, the nature of the constituents provided by the grammar is completely orthogonal to the question of incremental interpretation.

3.1 Context-free grammars

Given a set W of *words* (or *terminal symbols*), a context-free grammar over W is determined by specifying

1. a set N of *non-terminal* symbols
2. a set P of *productions* of the form $A \Rightarrow w_0 B_1 w_1 \dots w_{n-1} B_n w_n$

$$\frac{\pi \in P}{\vdash_G \pi : \tau(\pi)}\ \textsc{Con} \qquad \frac{\Gamma \vdash_G M : \alpha \to \beta \qquad \Delta \vdash_G N : \alpha}{\Gamma, \Delta \vdash_G MN : \beta}\ {\to}\text{E}$$

$$\frac{\Gamma, x : \alpha \vdash_G M : \beta}{\Gamma \vdash_G \lambda x.M : \alpha \to \beta}\ {\to}\text{I} \qquad \frac{}{x : \alpha \vdash_G x : \alpha}\ \textsc{Ax}$$

Figure 1: typing rules for the linear λ-calculus

The language of a context-free grammar G can be characterized in many equivalent ways (see e.g. Autebert et al. [1997]), with perhaps the most familiar one being in terms of rewriting.

A *sentential form* is a string of terminals and non-terminals. Given a sentential form $\alpha A \beta$, and a production $A \Rightarrow \gamma$, we say that "$\alpha\gamma\beta$ derives in one step from $\alpha A \beta$" and write $\alpha A \beta \Rightarrow_G \alpha\gamma\beta$. The language of a non-terminal A is the set of all terminal strings w which can be derived from A in any number of steps: $L_G(A) := \{w \in W^* : A \Rightarrow_G^* w\}$, and the language of the grammar is the language of the designated non-terminal S: $L(G) := L_G(S)$.

Another useful (and equivalent) characterization of the language of a context-free grammar is the following [de Groote, 2001]. A production $\pi = A \Rightarrow \mathtt{a}B\mathtt{b}C\mathtt{c}$ mixes two sorts of information, which are usefully distinguished: 1. the structural fact that π spawns two sub-derivations (of categories B and C respectively) 2. the phonological fact that the pronunciation of the derivation of category A rewritten by π consists of the letter $\mathtt{a}$ followed by some string of category B followed by $\mathtt{b}$ followed by a string of category C followed by $\mathtt{c}$. To each production $\pi = A \Rightarrow \alpha$ we associate its *syntactic type*; $\tau(\pi) := \mathsf{type}(\alpha, A)$, where

$$\begin{aligned} \mathsf{type}(\epsilon, A) &= A \\ \mathsf{type}(\mathtt{a}\alpha, A) &= \mathsf{type}(\alpha, A) \\ \mathsf{type}(B\alpha, A) &= B \to \mathsf{type}(\alpha, A) \end{aligned}$$

Thus the syntactic type of the production $\pi = A \Rightarrow \mathtt{a}B\mathtt{b}C\mathtt{c}$ is $B \to C \to A$. This production π can be *interpreted* as an action on *strings* in the following manner: $\mathcal{L}(\pi) = \lambda x, y.\mathtt{a}x\mathtt{b}y\mathtt{c}$. The λ-term $\mathcal{L}(\pi)$, of type $\mathbf{str} \to \mathbf{str} \to \mathbf{str}$ (with $\mathbf{str}$ the type of strings), encodes the phonological information encoded in the production $\pi = A \Rightarrow \mathtt{a}B\mathtt{b}C\mathtt{c}$.

The set of syntactic terms of type S are those for which the judgment $\vdash_G M : S$ can be derived using the inference rules in figure 1. A judgment of the form $\Gamma \vdash_G M : \alpha$ asserts that the λ-term M has type α in the context

Γ; here a context is a function assigning types to the free variables in the term M. The notation $x : \alpha$ denotes the context which assigns just the type α to just the variable x. Contexts Γ and Δ can be combined (written Γ, Δ) just in case they deal with different variables. These inference rules simply encode in a proof-theoretic format that (Con) constants have the types that they have, (Ax) a variable x has the type α in a context which assigns the type α to the variable x, ($\rightarrow$E) a function of type $\alpha \rightarrow \beta$ must be applied to an argument of type α and gives a result of type β, and ($\rightarrow$I) abstracting over a variable of type α in an expression of type β results in a function of type $\alpha \rightarrow \beta$. In a context-free grammar, we only ever need rules Con and $\rightarrow$E to type the syntactic terms of type S.

The set of strings generated by a context-free grammar is the phonological interpretations of the syntactic terms of type S: $L(G) = \{\hat{\mathcal{L}}(M) : \vdash_G M : S\}$, where for any map f from constants to λ-terms, $\hat{f}$ is its homomorphic extension over λ-terms as follows:

$$\begin{aligned}\hat{f}(x) &= x\\ \hat{f}(\pi) &= f(\pi)\\ \hat{f}(MN) &= \hat{f}(M)\hat{f}(N)\\ \hat{f}(\lambda x.M) &= \lambda x.\hat{f}(M)\end{aligned}$$

3.2 Parsing

Parsing is usefully considered as a *search* problem [Newell and Simon, 1976]: one searches through the structures defined by the grammar looking for those which match the input. There are not only different strategies for navigating through the search space (breadth-first, depth-first, A^*,...), but also different strategies for *constructing the search space* from the grammar (top-down, bottom-up, left-corner,...). All of these variations are well understood [Grune and Jacobs, 2008, Kallmeyer, 2010].

Consider the grammar with one non-terminal S and two productions 1. $\pi_1 := S \Rightarrow \mathtt{a}S\mathtt{b}S$ and 2. $\pi_2 := S \Rightarrow \epsilon$. Here, $\tau(\pi_1) = S \rightarrow S \rightarrow S$ and $\tau(\pi_2) = S$. $\mathcal{L}(\pi_1) = \lambda x, y.\mathtt{a}x\mathtt{b}y$ and $\mathcal{L}(\pi_2) = \epsilon$. The well-formed terms of type S are π_2, and $\pi_1\ \rho_1\ \rho_2$, for ρ_1 and ρ_2 well-formed terms of type S. The string `aabb` is the phonological interpretation of the term $\pi_1\ (\pi_1\ \pi_2\ \pi_2)\ \pi_2$. A top-down parse of the sentence `aabb` constructs this term in a left-to-right way, as shown in figure 2. The partial trees which represent the parser's current

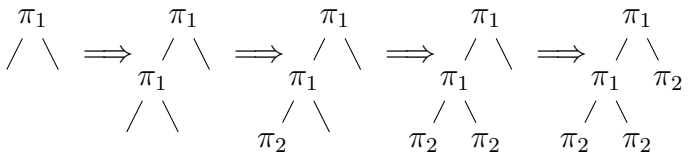


Figure 2: Stages of a top-down parse of *aabb*

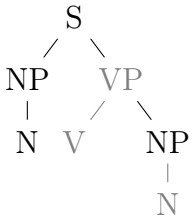
guess about the derivation of the sentence being parsed have an ontologically perfectly respectable characterization as λ-terms, as in the table below.

parse	**predicted string**
$\lambda x, y.\pi_1\ x\ y$	$\lambda x, y.\mathtt{a}x\mathtt{b}y$
$\lambda u, v, y.\pi_1\ (\pi_1\ u\ v)\ y$	$\lambda u, v, y.\mathtt{aa}u\mathtt{b}v\mathtt{b}y$
$\lambda v, y.\pi_1\ (\pi_1\ \pi_2\ v)\ y$	$\lambda v, y.\mathtt{aab}v\mathtt{b}y$
$\lambda y.\pi_1\ (\pi_1\ \pi_2\ \pi_2)\ y$	$\lambda y.\mathtt{aabb}y$
$\pi_1\ (\pi_1\ \pi_2\ \pi_2)\ \pi_2$	$\mathtt{aabb}$

The structures constructed incrementally by a *Dynamic Syntax* analysis are related to the structures incrementally assembled by a top-down parser. In particular, leaf nodes with question marks in their label play a role similar to the unverified predictions of a parser (the branches leading to nothing in the above).

3.3 Semantics

A compositional semantics is a homomorphism $[\![\cdot]\!]$ mapping abstract structures (derivations) to semantic values. It is convenient to treat semantic values as mediated through a representation; closed λ-terms up to $\beta\eta$-equivalence are a reasonable choice. Note that if you have an incomplete derivation, this can be represented as a λ-term of higher type (as shown in the previous section), and can be associated with a regular semantic value in a completely regular way. Thus, if you have a compositional semantics, and you are interpreting a term $K \equiv MN$, then $[\![K]\!] \equiv [\![M]\!]\,[\![N]\!]$. This means in particular, that for *any way* of decomposing a structure into parts of *any kind*, those parts can be semantically interpreted and the interpretations combined (via application) to give rise to the correct meaning of the original structure. As a concrete example, consider the tree in figure 3, which I wish to break into the two (discontinuous!) parts indicated in the figure by shading. As



$$\begin{aligned} &\mathsf{s}\ (\mathsf{np}\ \mathsf{n})\ (\mathsf{vp}\ \mathsf{v}\ (\mathsf{np}\ \mathsf{n})) && \text{(as a term)}\\ &\lambda F.\mathsf{s}\ (\mathsf{np}\ \mathsf{n})\ (F\ (\lambda x.\mathsf{np}\ x)) && \text{(one part)}\\ &\lambda f.\mathsf{vp}\ \mathsf{v}\ (f\ \mathsf{n}) && \text{(the other part)} \end{aligned}$$

Figure 3: A tree and its parts

a λ-homomorphism, the compositional semantics $[\![\cdot]\!]$ distributes down to the constants in a term, and therefore we have the equivalence below.[5]

$$\begin{aligned} &[\![\mathsf{s}\ (\mathsf{np}\ \ \mathsf{n})\ (\mathsf{vp}\ \mathsf{v}\ (\mathsf{np}\ \mathsf{n}))]\!]\\ &\quad = [\![\mathsf{s}]\!]\ ([\![\mathsf{np}]\!]\ [\![\mathsf{n}]\!])\ ([\![\mathsf{vp}]\!]\ [\![\mathsf{v}]\!]\ ([\![\mathsf{np}]\!]\ [\![\mathsf{n}]\!]))\\ &\quad \equiv (\lambda F.[\![\mathsf{s}]\!]\ ([\![\mathsf{np}]\!]\ [\![\mathsf{n}]\!])\ (F\ (\lambda x.[\![\mathsf{np}]\!]\ x)))\ (\lambda f.[\![\mathsf{vp}]\!]\ [\![\mathsf{v}]\!]\ (f\ \ [\![\mathsf{n}]\!]))\\ &\quad = [\![\lambda F.\mathsf{s}\ (\mathsf{np}\ \mathsf{n})\ (F\ (\lambda x.\mathsf{np}\ x))]\!]\ [\![\lambda f.\mathsf{vp}\ \mathsf{v}\ (f\ \mathsf{n})]\!] \end{aligned}$$

What should be concluded from this is that, if you have a compositional semantics, then online incremental interpretation comes for free.[6] As shown by de Groote [2006] (see especially Lebedeva [2012]), dynamic phenomena are easily and elegantly situated in a homomorphic perspective.

4 Incomplete Sentences

KCGC claim that the fact that "in conversation, commonly, we do not speak in complete sentences" "undermines basic theoretical notions like the abstract and folk-linguistic concepts of 'sentence' and any assumptions about string-level constituency." These "basic theoretical notions," they claim, are intrinsic to "standard models" of grammar, which are those maintaining the competence-performance and levels distinctions.

KCGC give examples of people finishing each other's sentences, such as 3 (adapted from their example 3).

[5] Instead of writing $[\![\cdot]\!]$ (the homomorphic extension of $[\![\cdot]\!]$), I write again $[\![\cdot]\!]$, thereby sacrificing notational distinctiveness for concision.

[6] There is no need, as KCGC worry in section 3.2, that the strict competence hypothesis might thereby be violated.

3. (a) Did Jo...

   (b) stumble?

What is the problem supposed to be here? I believe it stems from the assumption that 'standard models' of grammar require that each utterance of each discourse participant be completely analysable in isolation from any other. Thus, the question KCGC seem to be asking is: *what is the structure of 3b (or 3a)*?

I reject this assumption, and believe that this question may not even be meaningful. In the next sections, I explain this remark.

4.1 What is going on in split utterances?

KCGC propose that the split utterance in 3 should be analyzed as the hearer **b** simply finishing **a**'s sentence. This seems completely reasonable to me.[7] In this section I sketch how to achieve this in the context of a standard grammatical framework.

My story is based on the idea that hearers are able to guess, sometimes, what speakers are about to say. In other words, a hearer has some expectations about what might be coming next in a speech situation. (Or more generally, in everyday life.) Upon hearing words $u = w_1 \dots w_i$, a rational hearer conditions their expectations about what will come next on the fact that the string the speaker is producing begins with u. The hearer can then, instead of waiting to hear more, produce a most likely continuation of u (whether it be the next word, two words, phrase, etc). From this perspective, the dialogue in 3 is to be analyzed as the speaker **a** attempting to utter a sentence beginning with *Did Jo*, then being interrupted by **b**, who, anticipating that the sentence having been parsed will continue with *stumble*, chooses to vocalize this hypothesis. While there are many ways to interpret questions about *the structure of 3b*, the question KCGC ask is based on a presupposition (that 3b is really somehow a sentence in disguise) that is not met.[8]

A common way of formalizing this idea makes use of a language model;

[7] I find most of KCGC's analyses eminently reasonable.

[8] On the other hand, it really *is* a part of a sentence, the utterance of which was just distributed across multiple speakers, all of whom are parsing it.

a probability distribution over strings (and/or trees).[9] A particularly simple language model is obtained by adding *weights* to the rules of one's grammar.[10]

For concreteness, I summarize this simplistic model of discourses (which I treat here as strings of words with tags like "speaker 1:" and "speaker 2:" strewn about) in algorithm 1. I take as given a collection of probabilistic grammars, representing the language models of discourse participants. The

Algorithm 1 Discourse modeling

```
1:  speaker ← CHOOSEFIRSTSPEAKER(num_speakers)
2:  discourse ← PRINTSPEAKER(speaker)
3:  sentence ← ϵ
4:  done? ← false
5:  repeat
6:      w ← NEXTWORD(LangModel_speaker, sentence)
7:      if w = "." then
8:          done? ← DECIDEIFDONE
9:          sentence ← ϵ
10:     else
11:         discourse ← discourse ++ w ++ " "
12:         sentence ← sentence ++ w ++ " "
13:         if DECIDEIFNEWSPEAKER then
14:             speaker ← CHOOSENEXTSPEAKER(num_speakers, speaker)
15:             discourse ← discourse ++ PRINTSPEAKER(speaker)
16: until done? return discourse
```

narrowly linguistic part of the model is line 6, where the function NEXTWORD is called. This function computes the most likely next word based on the current speaker's language model, and the sentence spoken thus far. In context-free grammars and related formalisms (such as tree adjoining and minimalist grammars), this can be done efficiently [Nederhof and Satta, 2014].

Algorithm 1 makes use of operations such as DECIDEIFNEWSPEAKER and CHOOSENEXTSPEAKER to choose who utters the next word. A simple instantiation flips a coin to decide if there should be a new speaker, and rolls a

[9] This is not to commit to one position or another in a debate about whether grammar is probabilistic. Instead, a language model can be thought of as a high level picture on how experience might shape language use [Crocker, 2005]. (See also Chater et al. [1998].)

[10] In order to define a proper probability distribution, these weights need to satisfy certain conditions [Grenander, 1967].

die to choose who it should be. This can only be a poor approximation; there are certainly generalizations about when interruptions occur, and by whom. These however involve plans, goals, intentions, and many other things that my colleagues in the other cognitive sciences have much more knowledge about than do I. As a possible generalization which involves grammar, I could imagine that discourse participants who are surer about the next word are more likely to attempt to finish the speaker's sentence. Implementing this would require supplying the operations DECIDEIFNEWSPEAKER and CHOOSENEXTSPEAKER with an extra argument representing the sentence thus far, on which basis the relative confidence of the discourse participants in the next word could be computed.

4.2 Globally unlicensed derivations

Whatever merits algorithm 1 might have, it presupposes that all sentences throughout the discourse, split or not, are licensed by the grammar of each speaker. KCGC suggest that this is *not* in fact representative of reality.

> there is no indication that, either in monologue or dialogue, the continuation to some initial sequence must have been planned well in advance at the sentential/propositional level so that appropriate globally licensing derivations can be invoked by the parser/generator.

Example 4 (their (65)) gives a concrete example of this.

4. **Mary** Did you burn

   Bob myself?

5. $*$ Did you burn myself?

Assuming that neither speaker's grammar produces 5, then the discourse in 4 will not be generated by algorithm 1.

The straightforward resolution to this puzzle, pursued by KCGC (see my footnote 7) is to make the objects the speakers and hearers are generating more abstract. In other words, both Mary and Bob are still parsing/constructing the *same* syntactic structure (as in the previous section), it just gets pronounced in different ways.

One natural approach has it that the listener predicts what the speaker intends at a semantic level, and then generates based on this semantic representation what next to say. This in fact seems to be close to the strategy of KCGC. (For the upcoming discussion, it will be useful to keep KCGC's example (66) in mind.) After Mary's partial utterance, Bob's top-down parse of the sentence (ignoring the subject-auxiliary inversion) has led him to construct the partial structure $\lambda x.\mathsf{s}\ (\mathsf{np\ you})\ (\mathsf{vp\ burnt}\ x)$, the semantic import of which (in the context where $\mathbf{you} \mapsto \mathbf{bob}$) is $\lambda x.(x\ \mathbf{burn})\ \mathbf{bob}$. At this point, Bob must be able to 1. predict that **bob** is the intended semantic argument of the predicate, and 2. create a production plan which allows him to achieve this. Both of these tasks are far from being solved, however I recast them in concrete terms for the sake of discussion. Bob must find a syntactic expression M of type $\mathtt{np}$ (the type of the syntactic variable x), where $[\![M]\!]$ is of semantic type $(e \to t) \to t$ (the type of the semantic variable x), such that $(\lambda x.x\ \mathbf{burn\ bob})\ [\![M]\!] \equiv \mathbf{burn\ bob\ bob}$. For the first task, if Bob had a prior over λ-terms (at each type), he could, after conditioning his prior so that terms of type t match $(\lambda x.x\ \mathbf{burn\ bob})\ \bullet$, sample from the posterior, and use the subterm matching $\bullet$ to identify its possible linguistic realizers.[11] There are three lexical possibilities (reflexive, pronoun, or proper name), of which, only the first person reflexive is appropriate. Specifying things further will depend on how the binding theory is implemented. KCGC's strategy, recast in the present terms, requires us to consider the internal structure of our semantic λ-terms.

5 Conclusion

The program of *Dynamic Syntax* encourages formal linguists to engage with less rarified data than has typically been the case. It has been, after all, the great gambit of generative linguistics to assume that there is in fact a genuine *linguistic* causal power to study, as opposed to the possibility that language behaviour is not a natural class of phenomena after all. One would hope that the study of a genuine causal power would lend itself to better understand the phenomena in the world that it participates in!

KCGC claim that the methodologies of the competence-performance distinction, of levels of analysis, and of modularity are incompatible with this

[11] Kanazawa [2007] shows that exact generation is efficiently computable in context-free formalisms.

more naturalistic data. They claim that the grammar framework itself must build structures incrementally from left-to-right. These issues are unsupported by any valid argument and (I believe) deeply confused; more importantly, they are completely orthogonal to the matter at hand.

What is actually relevant are 1. the rejection of the assumption that each utterance must have a complete analysis in isolation of any context of use, and 2. the idea that some phenomena of language use are best made sense of by making reference to the way that lingusitic knowledge is used. I can hardly imagine that these two points should be very contentious.

Like all grammar formalisms, *Dynamic Syntax* makes some analyses easier to state, some harder (maybe even impossible), and shapes the analyses of its practicioners in sometimes novel ways. The more perspectives on some phenomenon we have, the better we are able to see past the accidental and into the essential. It often turns out that the real substantive issues are not those that were originally suspected. By clarifying the assumptions of different frameworks, we can begin to understand what they are really telling us about the world.

References

John R. Anderson. *The adaptive character of thought*. Lawrence Erlbaum, 1990.

Jean-Michel Autebert, Jean Berstel, and Luc Boasson. Context-free languages and push-down automata. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 1, chapter 3, pages 111–174. Springer, 1997.

Nick Chater, Matthew J. Crocker, and Martin J. Pickering. The rational analysis of inquiry: the case of parsing. In M. Oaksford and N. Chater, editors, *Rational Models of Cognition*, chapter 20, pages 441–468. Oxford University Press, 1998.

Noam Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, 1956.

Noam Chomsky. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, Massachusetts, 1965.

Matthew W. Crocker. Rational models of comprehension: Addressing the performance paradox. In A. Cutler, editor, *Twenty-First Century Psycholinguistics: Four Cornerstones*, chapter 22, pages 363–380. Lawrence Erlbaum, 2005.

Philippe de Groote. Towards abstract categorial grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, pages 148–155, 2001.

Philippe de Groote. Towards a montagovian account of dynamics. In M. Gibson and J. Howell, editors, *Proceedings of SALT 16*, pages 1–16, 2006.

Ulf Grenander. Syntax-controlled probabilities. Technical report, Brown University, Providence, RI, 1967.

Dick Grune and Ceriel J. H. Jacobs. *Parsing Techniques*. Springer, second edition, 2008.

Riny Huybregts. The weak inadequacy of context-free phrase structure grammar. In G. de Haan, M. Trommelen, and W. Zonneveld, editors, *Van Periferie naar Kern*, pages 81–99. Foris, Dordrecht, 1985.

Aravind K. Joshi, Leon S. Levy, and Masako Takahashi. Tree adjunct grammars. *J. Comput. Syst. Sci.*, 10:136–163, 1975.

Laura Kallmeyer. *Parsing beyond context-free grammars*. Springer, 2010.

Makoto Kanazawa. Parsing and generation as datalog queries. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pages 176–183, Prague, 2007. Association for Computational Linguistics.

Ekaterina Lebedeva. *Expression de la dynamique du discours à l'aide de continuations*. PhD thesis, Université de Lorraine, 2012.

David Marr. Artificial intelligence – a personal view. *Artificial Intelligence*, 9(1):37–48, 1977.

David Marr. *Vision*. W. H. Freeman and Company, New York, 1982.

Mark-Jan Nederhof and Giorgio Satta. Prefix probabilities for linear context-free rewriting systems. *Journal of Logic and Computation*, 24(2):331–350, 2014.

Allen Newell and Herbert A. Simon. Computer science as empirical inquiry: symbols and search. *Communications of the ACM*, 19(3):113–126, March 1976.

Stuart M. Shieber. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343, 1985.

Edward P. Stabler. Derivational minimalism. In C. Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer-Verlag, Berlin, 1997.

Nikolaas Tinbergen. On aims and methods of ethology. *Zeitschrift für Tierpsychologie*, 20:410–433, 1963.