J. Linguistics **00** (0000) 1–11. © 0000 Cambridge University Press
doi:10.1017/S0000000000000000 Printed in the United Kingdom

Remarks on Minimalist Feature Theory and Move[1]

DAVID ADGER

Queen Mary, University of London

(Received)

1. INTRODUCTION

I'd like to start this brief response to Asudeh and Toivonen's review article 'Symptomatic Imperfections' (hence SI) with a word of thanks for the careful job they've done and generally collegiate manner in which they've done it. They make many good points about Adger (2003) (Core Syntax—hence CS) which will help a great deal in improving the book for any future edition. Because of this, I don't want to dwell on any minor quibbles I have about their comments. Instead, I'd like to focus on some broader conclusions they draw about the theoretical devices used in CS and the perspective taken there. My remarks will address mainly SI's concerns about the feature theory I outlined in CS, and the question of movement.

2. FEATURE THEORY

SI claims (section 4.3) that there is no widely adopted formalization of feature theory in the Minimalist literature, but rather there are 'putatively intuitive understandings of features and feature checking'. They point out, correctly, that the formalizations of minimalism given by Stabler (1998), Retoré & Stabler (2004),

Vermaat (2004) and others using Linear Logic and Multi-Modal Categorial Grammar (see their footnote 8) have not found their way into the mainstream literature. Their claim is true of CS and much of the mainstream literature, but I think it is beside the point. Firstly, the informal presentation I gave in CS, which follows Chomsky (1995, 2000, 2001) fairly faithfully, maps rather directly onto Stabler's formalization; that is, for the most part there is a one-to-one correspondence between the concepts I informally present and Stabler's formalization, as I think is fairly clear from a reading of CS with Stabler's work in mind. This is unsurprising, as both Stabler and I essentially use the notions developed by Chomsky.

Secondly, and here is where the difference between the authors of SI and myself probably lies, I think it is perfectly legitimate to do (theoretical) linguistics in the absence of a formalization and that one can gain a great deal of insight in this way. Once some theoretical understanding of the phenomenon is developed, then one can ask what a good formalization of the theory is and when the theory changes, one can see whether the underlying formalization needs to be revised (extended or reduced). I think that things go the other way around for the authors of SI: the underlying formalism constrains the theory in which an analysis can be proposed, so one needs a formalism to do linguistics; however, even under this view, the formalism can be extended in various ways if necessary to ensure that an analysis of a linguistic phenomenon will comport with it (e.g. the addition of set-valued features to LFG in Kaplan & Bresnan 1982, or the use of functional uncertainty in LFG by Kaplan & Zaenen 1989, or the addition of boolean constraints on feature values in HPSG (Pollard & Sag 1994)). In the end, I think both of these viewpoints come to pretty much the same thing. You have a theory with a formalization, and

if the theory needs to be changed, you change it, and the formalization changes with it.

The remaining concerns that SI has about the feature system in CS are: it is (i) non-compositional, (ii) complex, and (iii) unconstrained. In what follows I will show that it is (i′) compositional, (ii′) simpler than the recursive attribute-value systems used in LFG/HPSG and (iii′) fairly constrained and, in principle, further constrainable (to the same extent that any feature system is).

2.1 *Compositionality*

In chapter 2 of core syntax, I introduce three different feature systems to the reader: a privative system, a binary system and an attribute-value system. The purpose of doing this is really so that students know the range of possibilities, and I therefore try to provide motivations for all of these systems. I do indeed consider the privative system to be the simplest (see the next section), but I'm not sure that it's empirically tenable. However, for the moment, I'll put this aside, and address the question of whether the system I provide is compositional or not. SI claims that it is not on the basis of the following quote from CS:

> Words which are specified just as [singular] are singular morphologically and are interpreted semantically as single entities. Words which are syntactically [plural] have plural morphology and are interpreted as referring to a group of entities. Words which have the feature specification [singular, plural] are dual in form and are interpreted as referring to pairs of entities. (p28)

From this, SI concludes "It is clear from this quote that [singular, plural] is not to be interpreted such that its interpretation is made up of the interpretation of its parts". I think that it's not so clear, and this was certainly not my intention. I believe very strongly in compositionality in the syntax/semantics computation, and have consistently argued that grammatical features are compositionally interpreted since my thesis (Adger 1994). I think that the misunderstanding, for which I am partly responsible, arises from the fact that I give imprecise definitions of the meanings of these features in this quote. However, I do note, in my discussion of the interpretation of [plural] via an interface rule three pages later (p31), that this rule is not 'correct as stated' and that 'it needs to be much more carefully worked out (so that the semantics of dual forms is correct, for example)'.

Let me do this now. Following, but adapting, Harbour 2003, I will take the features [singular] and [plural] to be given the following interpretations (N can be taken to be the set of elements in the denotation of the common noun on which the feature is instantiated; it is assumed that $N \neq \emptyset$—see Harbour's work for discussion). Writing $[\text{F}]_N$ for the feature [F] holds of N, we have :

(1) $[\text{singular, A}]_N$ if, and only if, $\neg\exists N' \subset N$, $[\text{A}]_{N'}$

(2) $[\text{plural}]_N$ if, and only if, $|N| \neq 1$

The interpretation of the feature [singular] essentially guarantees that it is impossible to remove an element from N, leaving behind a non-empty set, without

making a substantial difference to the (number) properties of the whole thing. See Harbour's work for an extended defense of features with these kinds of meanings.[2]

We can then prove that the semantics of [singular, plural] will give a set with cardinality 2. I show this here, following Harbour's proof. Harbour also gives proofs that the system gives the right result for [singular] being interpreted as 1 and [plural] as 3 or more. The important idea here is that we can give a well motivated semantics for number features from which it follows that the descriptive category "dual" is the result of a combination of the features used in the analysis of singular and plural, just as I said in the quote above from CS.

(3) $[\text{singular, plural}]_N$ iff $\neg\, \exists\, N' \subset N$, $[\text{plural}]_{N'}$

iff $\forall N' \subset N\ \neg\ [\text{plural}]_{N'}$

iff $\forall N' \subset N, \neg\, |N'| \neq 1$

iff $\forall N' \subset N, |N'| = 1$

iff $|N| = 2$

SI also points out other putative problems with the analysis of Hopi number that motivated my suggestion that dual was [singular, plural], which I don't have space to deal with here, but the crucial claim they are making is that the feature system I assume "ALLOWS NONCOMPOSITIONAL FEATURE COMBINATION AT ALL" (their caps). But nowhere do I say this, and in fact I state that, via interface rules "a particular feature gives rise to ... a particular interpretation" (CS, p32). Together with the claim that "Words which have the feature specification [singular, plural] are dual in form and are interpreted as referring to pairs of entities", I think it

[2] Harbour's system actually adopts binary features here, and my [singular] corresponds with his [−augmented] while my [plural] corresponds with his [−singular]. However, this is irrelevant for my purposes here, which are just to show how to give a compositional interpretation of privative number features.

follows that I take the interpretation of the dual category to be derived from that of the singular and plural. The proof above shows one way of doing this; see Harbour's work for a detailed defense of the idea that various descriptive number categories (paucal, trial etc) can be derived via the composition of a small set of features.

Let me state this result clearly: In the system developed in CS, each (interpretable) feature has an interpretation established by an interface rule. When these features occur together on a single lexical item, they are interpreted by composing the meanings (not explicitly stated in CS, but not denied, and something which is a reasonable inference from things which are explicitly stated). The precise mode of composition is determined by the meanings of the features and by rules of interpretation for complex expressions (these might be type driven, or stipulated).

One final point on this: SI provides an alternative compositional analysis, which appeals to the interpretation of number features on a verb, versus those on an argument. This analysis is actually impossible in the CS system as it uses meanings of uninterpretable features—in CS uninterpretable features do not have meanings. So the CS approach is in fact more restrictive, as well as being compositional.

2.2 *Simplicity*

The second criticism that SI has of the feature system in CS is that it mixes privative and attribute-value features. SI says 'a mixed system is arguably not as simple as a uniform system that uses only attribute-value features'. SI also suggests that such a system might be harder to acquire. These points are well taken; I would indeed prefer a system that is only privative, because I think that

such a system, which I've just shown can be compositional, is more elegant. However, I do allow a mixed system with privative features, binary features and attribute-value features. Note that the complexity stops there, though. I do not allow recursive attribute value structures of the kind found in LFG F-structures or HPSG AVMs. This is actually because I consider these to be considerably more complex.

Let me show how. Let us take a privative system. In such a system, a lexical item is a set of features; we can write a simple grammar for this:

(4) a. LI $\rightarrow \{F_1, ..., F_n\}$

where each F is an atomic symbol.

Now, if we have a binary system, we add a rule expanding F_i with the following (assume for concreteness that the attributes are a, b, c and d):

(5) a. $F_i \rightarrow$ [Att:Val]

b. a, b, c, d are atomic symbols drawn from Att, the set of attributes

c. $+, -$ are atomic symbols drawn from Val = $\{+, -\}$

A non recursive attribute value system is of broadly the same level of complexity, but we need to index values to the attributes they are values of. We can do this using context sensitive rules:

(6) a. Val $\rightarrow$ x / a:__

b. Val $\rightarrow$ y / a:__

c. Val $\rightarrow$ y / c:__

d. Val $\rightarrow$ z / d:__

Here, x, y and z are atomic symbols drawn from Val. This grammar states that x and y are possible values of a, that y is also a possible value of c, and that z is a possible value of d.

Now, however, let us think what we need if we are to have recursive attribute value structures. The grammar itself is trivial, we simply change our value rule to read:

(7) Val $\rightarrow F_i^+$

This has the effect of allowing an attribute to be a possible value of another attribute, giving:[3]

(8) [a:[b:[c:[d:z]]]]

But now there is simply no straightforward way of ensuring that values can be restricted to particular attributes (what SI refers to as 'typing' the features). We can't write a context sensitive rule to do this, since non-adjacent information is required: Val expands as F, and then F as [Att:Val], so there is no possible context sensitive rule which allows us to say that an attribute, which is a value of a higher attribute, can be keyed to the category of that higher attribute, as the higher attribute is never its sister.

The solution is then either to allow unrestricted rewrite rules, or to type the features outside the formal feature system itself, and go with a more powerful feature theory. HPSG does this by adopting a hierarchy of types, and LFG does it by defining features as functions, rather than as formulae, and by explicitly stating which functions can be arguments of which others. These are perfectly reasonable

[3] The cross here allows one or more occurrences of the F so we can model branching AVMs ([a:$[^{b}_{c}]$]). I ignore this extra complication here.

solutions, but they require a theory beyond what is expressible via these simple feature grammars. Moreover, both LFG and HPSG allow much more complexity than this: LFG allows set valued functions and lattice theoretic interpretations of functional labels, while HPSG's AVMs include ordered sequences, boolean operations, etc. The system in CS does not.

A system which does not allow feature recursion is, then, at least from this perspective on complexity, far simpler: the theory of lexical items and the theory of features proposed in CS is actually therefore markedly less complex than that proposed in the unification based lexicalist theories the authors of SI advocate, in contrast to what is claimed in SI.

A reviewer objects that AVMs are not trees but 'more complex Directed Acyclic Graphs (DAGs)', so providing a PS-grammar as a formalization is "hopelessly confused". But DAGS are in fact just generalizations of trees (a rooted tree is a special kind of DAG). That is, the kinds of objects that DAGs can, and in the theories concerned do, model are more complex than the types of objects that even context sensitive PS grammars can model. Which is precisely the issue: the feature system in CS is simpler than that in either LFG or HPSG.

I think it is an interesting research question to investigate to what extent the simplification of lexical items implied by this 'hierarchy' of feature systems (privative, binary, non-recursive attribute value, and recursive attribute value) is possible; because of this, I prefer to start at the 'lower end' of the complexity scale for lexical items and impute to them as little structure as possible. Ideally, lexical items would be composed of privative features. If Harbour 2003 is right, then this isn't enough to capture the complexities of at least some natural languages (the Kiowa number system being the case that Harbour uses), and binary features

are necessary. It would be good to stop at this point, but perhaps non-recursive attribute value features of the sort proposed in CS are necessary, and perhaps even recursive attribute-value features. To be honest, I don't know, but I'd rather see how simple a system we can get away with in the structure of lexical items. I chose a 'middle path' in CS, mainly for pedagogical reasons, but I see the right level of complexity of the feature theory as an empirical issue, and I'd rather explore the strongest hypotheses first.

A final point: I'm not arguing that overall a grammar incorporating a feature system like that in CS is a priori better than one which has recursive attribute value features. There is a question of trade-off here to consider as well. The richly structured lexical items of the models preferred by the authors of SI (see their section 5) have a great deal of complexity that Minimalists would generally treat as part of the syntax, and it's an open question about where one "puts" the structure that is surely necessary: the kind of implementation of Minimalism in CS, and the unification-based lexicalist systems advocated in SI, are pursuing different research intuitions which both, I think, could potentially lead to interesting insights about human language. I just don't think we understand enough to say whether the right theory will involve lexical items with recursive attribute-value structures, or syntactic structures with many functional heads. They each have their strengths and weaknesses, and it would be better for the field if we could appreciate that a little more.

2.3 *Constrainedness*

The final criticism that SI makes of the feature system in CS is that it is unconstrained. The particular issues that SI raises are (i) an analysis of number

agreement in English present tense and (ii) subject wh-questions. I will focus on the first of these for reasons of space, and as I think that the points I make carry across mutatis mutandis to the second case.

For (i), I suggest that the uninterpretable feature *u*Infl can have either *past* or *singular* as a value. In the feature grammar just sketched, that means we have the following kinds of statement:

(9) a. Val → past / *u*Infl:__

b. Val → singular / *u*Infl:__

SI suggests that it is "instructive to think about what such a claim would mean in a typed feature theory" and argue that my analysis needs to be constrained further. But it's not instructive: SI's argument assumes that I need to add something extra to capture feature typing, for example a separate sub-theory, as in HPSG; however, I don't, because the system I'm working with doesn't require this extra theoretical power. I take (9) to be facts which English acquirers learn.[4] I don't think that this is unreasonable and it seems pretty restricted to me.

Of course, one might then ask whether there is a theory of such facts; that is, is there a reason for having certain kinds of statements of the sort in (9) above, and not others? More concretely, can, say, *past* ever be a value of *case*? In CS, I assumed that there was no such extra constraining theory for the values of uninterpretable features, so *past* can indeed be a value of *case*. Although the system I offered does not preclude an extra sub-theory to constrain uninterpretable features, the framework of assumptions that Minimalism adopts makes such a

[4] Note further that the feature *u*Infl is uninterpretable, so what is learned is just a fact about (the syntactic determinants of) agreement morphology, and not about meaning (I take the values of interpretable features to at least be potentially restricted by their semantics—again see Harbour's work). The same point can be made about the *u*clause-type feature in subject wh-questions which is valued as either *wh* or as *Q*. These features are never interpreted.

move unattractive in the absence of plausible interface constraints in which such a theory could be grounded. The values of interpretable features, in contrast, are tightly constrained by the semantics of these features, and in this sense they are, in fact, effectively "typed" (that is, there is just no interface rule which provides a meaning for [number:past], a fact which is irrelevant for [*u*number:past]).

Moreover, I'm not wedded to these particular analyses, and they shouldn't be taken as representative of Minimalist theory (which stipulates a strict matching condition on Agree which I relaxed for just these cases). My intention was to show students how one could develop analyses given the system. Perhaps a better way to go for the agreement facts would be to split *u*Infl into separate features of tense, person, and number, and then to complicate the spellout-rules for these features (this would be more standard, I think, and would solve the problem of what to do with present tense *be*). I made a judgment call on this, preferring to make a couple of stipulations about the matching abilities of two features rather than introducing further features on T and a number of spellout-rules. Perhaps it was the wrong call, although I don't think so.[5]

Overall, then, the system as presented is pretty constrained.

In summary, the feature system presented in CS is *in fact* compositional, it is *less* complex than that assumed in unification based lexicalist grammars, and it is *in fact* constrained and *in principle* further constrainable (for example, the possible values of attributes of interpretable features could be constrained by the semantics of these features). Although I'm grateful to the authors of SI for making

[5] On subject wh-questions, draft versions of the book adopted a Pesetsky & Torrego (2001) approach, but this meant that I had to introduce a lot of theoretical apparatus which went further than I'd wanted to go, and so I decided to adopt what I thought (and still think) was a minor stipulation well within the bounds of the system.

me think harder about these issues, I don't accept their criticisms of the theory presented in CS.

3. MOVEMENT

SI ends with a suggestion that it would "make sense for minimalists to make the final transition to a unification based lexicalist framework which is adequately formalized ... and which is not encumbered by the additional transformational mechanism of Move/internal Merge". We have seen that minimalism is in fact adequately formalized (Stabler's work and others—although of course whether it is adequately formalized is a subjective judgment) and that it has a take on the structure of lexical items that is rather different from existing lexicalist frameworks, and which, I think, asks an interesting question about the distribution of information in the grammar.

As for the question of Movement, there are two potential issues here: (i) the fact that the system is derivational; (ii) the fact that the system has a particular mechanism for capturing "apparent displacement" dependencies. On the question of derivationality, prominent "minimalists" have different views and there is a healthy debate in the literature with some arguing for a representational system with no transformational operations, while others argue for a derivational system (see, Brody 1997, Chomsky 1989, Kayne 2004, Rizzi 1990, for different viewpoints). The fact that most analyses seem to be statable in either terms perhaps just means that we don't understand the issues deeply enough yet; but in such a situation, a debate is a better way to proceed than simply removing one of the options by a meta-theoretical fiat "that minimalists should make the

final transition to a unification based lexicalist framework". As Chomsky has long suggested, this question is probably in the end empirical.

On the question of Movement, I confess I'm at a loss to understand the point SI makes. We need something to capture, say, wh–dependencies. We can use Functional Uncertainty, SLASH-features, Movement etc. Because of the assumption that lexical items are as simple as possible, category valued SLASH features are out for minimalists. Functional Uncertainty is tied to the C-Structure to F-Structure architecture of LFG and the possibility of mapping between two data structures in the syntax; the system outlined in CS has only one data structure in the syntax, again ruling this out as a possibility. I really don't see what the problem is with adopting a fairly straightforward extension of the core Merge mechanism, which allows Merge of some element that has already been Merged.[6] Move might not be a conceptual necessity (but see footnote 6), any more than either Functional Uncertainty or SLASH is, but it's a perfectly reasonable hypothesis about the dependency forming nature of the grammar, and this is exactly how it is presented and motivated in CS. Moreover, there are different conceptions of movement, all of which are being actively explored and some of which have been formalized (e.g. Vermaat 2004 suggests seeing movement as the hypothesis formation step in a natural deduction logic). I think that SI overstates its case here, perhaps because the authors are a little riled by the regretable tendency of some minimalists to overstate theirs.

The argument I'm making here is not an argument about formal complexity: it is about whether a linguistic theory with the operation Move is worse than a

[6] I put aside here Chomsky's argument, criticized in SI, that a system which has Merge without Move (Internal Merge) is one which places an arbitrary restriction on the operation. I think this argument actually has more legs than SI gives it credit for, but that is best left aside here, as I don't report this argument in CS.

linguistic theory without this operation. Well, that depends on the whole of the theory. Removing Move from Minimalist theory is akin to removing Functional Composition from Combinatory Categorial Grammar, or Functional Uncertainty from LFG. You would end up with a different theory, which would require some other mechanism to capture the empirical data. Is LFG a worse theory without Functional Uncertainty? Sure, if you want LFG to capture unbounded dependencies. Is CCG better off without Functional Substitution? Again, yes, but you'll need another way to capture parasitic gaps. In fact, though, perhaps removing Move is actually more akin to removing feature-typing from HPSG or unification from these theories in general, given that Move is so deeply embedded in the architecture of the theory.

I think SI's statement that Minimalism should just get rid of Move really boils down to not much more than a statement that people doing Minimalism should just do HPSG or LFG instead. I feel a bit more secure about the future of the field with a different attitude: people should explore whatever research ideas they find interesting within whatever framework gives them an insight into the problems they are working on. We surely know that none of them are right, but some of them may lead to ideas that are less wrong.

The last line of SI is "transformational grammar is one hypothesis among many, not the null hypothesis, nor, perhaps, the minimal one." I agree, but I think that it is an interesting hypothesis, that it has led to a great deal of insight and understanding of human language, and that it is a theory which is still well worth teaching our students. I hope CS is of some use in doing this.

REFERENCES

Adger, D. (1994). *Functional heads and interpretation*. Ph.D. dissertation. University of Edinburgh, Centre for Cognitive Science.

Adger, D. (2003). *Core syntax*. Oxford: Oxford University Press.

Brody, M. (1997). Perfect chains. In Haegeman, L. (ed.), *Elements of grammar: Handbook of generative syntax*. Dordrecht: Kluwer. 139–167.

Chomsky, N. (1989). Some notes on the economy of derivation and representation. In Freidin, R. (ed.), *Principles and parameters in comparative grammar*. Cambridge, MA: MIT Press.

Chomsky, N. (1995). *The minimalist program*. Cambridge, MA: MIT Press.

Chomsky, N. (2000). Minimalist inquiries: the framework. In Martin, R., Michaels, D. & Uriagereka, J. (eds.), *Step by step: Essays on Minimalist syntax in honour of Howard Lasnik*. Cambridge, MA: MIT Press. 89–115.

Chomsky, N. (2001). Derivation by phase. In Kenstowicz, M. (ed.), *Ken Hale: A life in language*. Cambridge, MA: MIT Press. 1–52.

Harbour, D. (2003). *Elements of number theory*. Ph.D. dissertation. MIT (to appear, 2006, Kluwer).

Kaplan, R. & Bresnan, J. (1982). Lexical Functional Grammar: A formal system for grammatical representation. In Bresnan, J. (ed.), *The mental representation of grammatical relations*. Cambridge: MIT Press. 173–281.

Kaplan, R. & Zaenen, A. (1989). Long distance dependencies, constituent structure and functional uncertainty. In Baltin, M. & Kroch, A. (eds.), *Alternative conceptions of phrase structure*. Chicago: Chicago University Press. 173–281.

Kayne, R. S. (2004). Prepositions as probes. In Belletti, A. (ed.), *Structures and beyond*. Oxford: Oxford University Press. 192–212.

Pesetsky, D. & Torrego, E. (2001). T to C movement: causes and consequences. In Kenstowicz, M. (ed.), *Ken Hale: A life in language*. Cambridge, MA: MIT Press. 355–426.

Pollard, C. & Sag, I. (1994). *Head Driven Phrase Structure Grammar*. Stanford: CSLI.

Retoré, C. & Stabler, E. (2004). Generative grammars and resource logics. *Research on Language and Computation* **2**. 3–25.

Rizzi, L. (1990). *Relativized minimality*. Cambridge, MA: MIT Press.

Stabler, E. (1998). Acquiring languages with movement. *Syntax* **2**. 72–97.

Vermaat, W. (2004). The minimalist move operation in a deductive perspective. *Research on Language and Computation* **2**. 69 – 85.