An assessment of some Characteristic Features of the Minimalist Program and Optimality Theory

BY

Godfrey K. DANIEL, *Ph.D*
Department of Linguistics and Communication Studies
University of California
Oakland, California
United States

ABSTRACT

It is obvious that the hybrid model may substantially change our views on the properties ascribed to the generative and filtering device by traditional MP/OT. It is also that one can simplify current MP by expressing all language-specific statements (including the more recent effect-on-output conditions discussed above) by means of language-specific rankings of otherwise universal violable constraints. It is believed that a formal way of expressing MP and other views is by assuming that such heads may or may not have an epp-feature. Linguistics are of the opinion that adding an OT-evaluation to MP can be done in a very 'minimal' way, simply by introducing the idea that the output of the computational system is filtered in an optimality-theoretic fashion by means of a language-specific ranking of otherwise universal constraints. The introduction of such an optimality-theoretic evaluator eliminates the need for many other devices that are currently used in MP to capture cross-linguistic differences such as language-specific filters of the type in (16) and parameter settings. In view of the aforementioned, every writer should ensure that in their write-up Minimalism rules should be applied where by elimination of excessive complexity of principles is observed.

KEYWORDS: Minimalist Program, Characteristic, Features, Optimality Theory

Introduction

It is obvious that hybrid system is very paramount in every linguistics or write-up. For such it is necessary to briefly lay out the properties of MP and OT. The focus here will be on OT, given that MP is more extensively discussed in various other part of this volume. An important conclusion will be that, contrary to what the name suggests, OT-syntax resembles MP in that it is not a theory but a program in the sense that it implies neither a specific theory of the generative component nor of the evaluative module that evaluates the output of the system; it is a theory of constraint interaction that computes the predictions for any generator and set of postulated constraints. We will furthermore argue that the overall modeling of the syntactic module presupposed by MP is very similar to that proposed by OT-syntax. This will make it possible to develop a new program that incorporates certain basic assumptions and guiding intuitions from both MP and OT. Section 3 and 4 will illustrate two implementations of the hybrid program.

The Minimalist Program

Given that minimalism is not a theory but a program, which refers to a family of approaches that aim at reducing syntax/grammar to its absolute minimum, we will simply pick out one of the more familiar approaches for illustration, viz., the one developed by Chomsky (1995) and, especially, subsequent work. The overall structure of the model that has arisen since Chomsky (2000) is given in Figure 1. Below, we will briefly summarize some of the properties of this model that will be central to our concern.

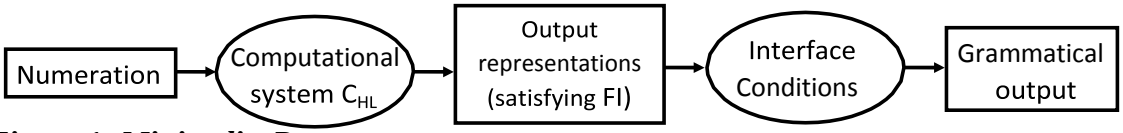

Figure 1: Minimalist Program

The derivation takes a numeration as its input, the elements of which are processed by the operations of the computational system for human language C_{HL}. The first operation is EXTERNAL MERGE (henceforth: Merge), which combines elements from the numeration and/or larger syntactic objects already formed into larger structures. The merged elements may contain unvalued formal features that must be valued by entering into the syntactic relation AGREE with some other element in their syntactic (c-command) domain with corresponding valued formal features: the unvalued features thus function as probes that search within a certain domain for a goal with corresponding valued features. It is further assumed that this probe can be assigned a so-called *EPP*-FEATURE, which requires that the goal be placed in its minimal domain (in the sense of Chomsky 1995: ch.3) by means of INTERNAL MERGE (henceforth: Move). When the numeration is exhausted, the subsequent applications of Merge and Move must have resulted in an output representation that satisfies Full Interpretation, that is, which only consists of elements that can be given an interpretation by either the Conceptual-Intentional (C-I) or the Articulatory-Perceptual (A-P) system; if not, the derivation crashes at these interfaces. The operations of C_{HL} are subject to LAST RESORT in the sense that they may only apply when forced: Merge must apply given that the derivation must result in a single syntactic object, which implies that the numeration must be exhausted at the end of the derivation; Agree is forced by Full Interpretation given that unvalued formal features cannot be interpreted by the C-I or A-P system. Move, finally, is forced by the need to eliminate the EPP-features: it is often assumed that these features must be eliminated immediately after they are introduced in the structure in order for the derivation to be able to proceed.

The computational system C_{HL} is seen as invariant among languages and defines a set of possible output representations for each numeration. The fact that languages vary in word order (that is, give rise to different output representations on the basis of similar numerations) is accounted for by assuming that languages may be parameterized with respect to the question whether a certain probe, like the unvalued formal feature(s) on the functional heads of the clause (including the light verb *v**), is associated with an EPP-feature.[3] In earlier minimalist work, it was assumed that the option of having or not having an EPP-feature was fixed once and for all in the lexicon of the language in question, but Chomsky (2001) suggested that the EPP-features can (at least sometimes) be optionally assigned to a certain probe, which may account for certain "optional" movements like object shift in Icelandic. However, given that object shift is sensitive to the information structure of the clause, Chomsky claims that (at least in this case) the assignment of an EPP-feature is subject to an effect-on-output condition: an EPP-feature can only be assigned when this has repercussions for the meaning of the clause. We will return to this in Section 3.

The effect-on-output condition, which in effect functions as a filter on the set of possible output representations, is an example of a larger set of so-called interface conditions (as were the global economy, bare output, interface, etc. conditions postulated in earlier and other versions of MP). Although these conditions are assumed to play an important role in selecting the grammatical output representations for a certain language L, it seems that the minimalist community has failed so far to develop a general format that such conditions must meet. The aim of this chapter is to show that OT may fill that gap.

1.1 Optimality Theory

This section will briefly discuss what we will refer to as traditional OT-syntax. Section 2.2.1 will start with presenting the basic ideas shared by researchers working within OT, which will subsequently be illustrated for syntax in Section 2.2.2. Section 2.2.3 will conclude this brief discussion of OT-syntax by showing that OT-syntax is not a theory, but instead resembles MP in that it functions as a research program.

1.2.1 Shared Basic Ideas

Just like MP, OT-syntax is not a theory but a program. It refers to a family of approaches that adopt the model of grammar in Figure 2. The guiding intuition of OT is that the language system consists of two components, viz., a generative device called GENERATOR that produces candidate sets and a language-specific filtering device called EVALUATOR that selects candidates from these candidate sets as optimal (well-formed) in a given language L.

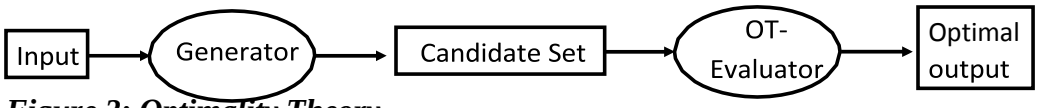

Figure 2: Optimality Theory

Furthermore, OT adopts the basic assumption that the evaluator consists of a universal set of VIOLABLE CONSTRAINTS (usually referred to as CON) and that it is the LANGUAGE-SPECIFIC RANKING of these constraints that determines which candidates from the candidate sets are optimal in L. The determination of the optimal candidate thus proceeds as in (1), which we have adapted from Archangeli (1997).

(1) The evaluator finds the candidate that *best satisfies* the ranked constraints, such that:

   a. violation of a lower ranked constraint is tolerated if this enables the candidate to satisfy a higher ranked constraint, and

   b. ties by violation or by satisfaction of a higher ranked constraint are resolved by a lower ranked constraint.

1.2.2 An Illustration

The way the OT-evaluator works can readily be demonstrated by means of Pesetsky's (1997;1998) analysis of the pronunciation patterns of relative clauses. Pesetsky adopts the standard assumption that relative clauses are derived by means of *wh*-movement of (the phrase containing) the relative pronoun, followed by optional deletion of the phonological content of the relative pronoun and/or the complementizer. His aim is to provide an OT- alternative for

Chomsky & Lasnik (1977) proposal that the pronunciation patterns in (2) arise as the result of the DOUBLY FILLED COMP FILTER and the RECOVERABILITY CONDITION on deletion.

(2) a. the man [who$_i$ ~~that~~ I know t_i] a□. the book [[$_{PP}$ about which]$_i$ ~~that~~ he spoke t_i]

b. the man [~~who~~$_i$ that I know t_i] b□. *the book [[$_{PP}$ ~~about~~ which]$_i$ that he spoke t_i]

c. the man [~~who~~$_i$ ~~that~~ I know t_i] c□. *the book [[$_{PP}$ ~~about~~ which]$_i$ ~~that~~ he spoke t_i]

d. *the man [who$_i$ that I know t_i] d□. *the book [[$_{PP}$ about which]$_i$ that he spoke t_i]

Pesetsky's proposal also aims at accounting for the fact that the pronunciation pattern is language-specific. The contrast between the primeless examples in (2) and (3) shows that English allows a wider range of pronunciation patterns with a bare relative pronoun than French. However, when the relative pronoun is embedded in a larger constituent, like the PPs in the primed examples, the two languages behave the same.

(3) a. *l'homme [qui$_i$ ~~que~~ je connais t_i] a□. l'homme [[$_{PP}$ avec qui]$_i$ ~~que~~ j'ai dansé t_i]

b. l'homme [~~qui~~$_i$ que je connais t_i] b□. *l'homme [[$_{PP}$ ~~avec~~ qui]$_i$ que j'ai dansé t_i]

c. *l'homme [~~qui~~$_i$ ~~que~~ je connais t_i] c□. *l'homme [[$_{PP}$ ~~avec~~ qui]$_i$ ~~que~~ j'ai dansé t_i]

d. *l'homme [qui$_i$ que je connais t_i] d□. *l'homme [[$_{PP}$ avec qui]$_i$ que j'ai dansé t_i]

Pesetsky accounts for the data in (2) and (3) by means of the universal constraints in (4), which we have slightly simplified here for reasons of exposition: (4a) is simply Chomsky & Lasnik's (1977) recoverability condition on deletion, (4b) is a constraint that favors embedded clauses introduced by a complementizer, and (4c) is a constraint that favors the deletion of function words (like complementizers).

(4) a. RECOVERABILITY (REC): a syntactic unit with semantic content must be pronounced unless it has a sufficiently local antecedent.

b. LEFT EDGE (CP): the first leftmost pronounced word in an embedded CP must be the complementizer.

c. TELEGRAPH (TEL): do not pronounce function words.

The analysis crucially relies on the fact that LE(CP) in (4b) and TEL in (4c) are in conflict: the former favors complementizers to be pronounced, whereas the latter favors them to be deleted. This makes it possible to account for variation between languages by varying the ranking of these constraints: when LE(CP) outranks TEL, as in (5a), we get a language in which embedded declarative clauses must be introduced by a complementizer; when TEL outranks LE(CP), as in (5b), we get a language in which embedded declarative clauses are not introduced by a complementizer; and when we assume that the two constraints are in a tie, as in (5c), we get a language in which embedded declarative clauses are optionally introduced by a complementizer.

(5) a. LE(CP) >> TEL: embedded declarative clauses are introduced by a complementizer.

b. TEL >> LE(CP): embedded declarative clauses are not introduced by a complementizer.

c. TEL < > LE(CP): embedded declarative clauses are optionally introduced by a complementizer.

It is important to realize that a tie like (5c) expresses that the rankings in (5a) and (5b) are *simultaneously* active in the language in question; the set of optimal candidates selected by (5c) is the union of the sets of optimal candidates selected by (5a) and (5b); see Müller (1999) for a discussion of various uses of the notion of tie. The evaluations can be made visible by means of tableaux. Tableau 1 gives the evaluation of embedded declarative clauses with and without a pronounced complementizer in a language with the ranking in (5a). The two asterisks indicate that the constraint in the header of their column is violated by the candidate in question. The (a)-candidate, with a pronounced complementizer, violates TEL but this is tolerated because it enables us to satisfy the higher ranked constraint LE(CP); cf. (1a). The (b)- candidate, with a deleted complementizer, violates LE(CP), and this is fatal, which is indicated by an exclamation mark, because the (a)-candidate does not violate this constraint. The (a)- candidate is therefore optimal, which is indicated by means of the pointed finger: □. The shading of cells following the fatal constraint violation indicates that these cells do not play a role in the evaluation; this convention is mainly for convenience, because it makes larger tableaux easier to read.

Tableau 1: ***No C-deletion in embedded declarative clauses***

	LE(CP)	TEL
a. □ .... [complementizer....]		*
b. .... [~~complementizer~~....]	*!	

Now consider the evaluation of the same candidates in a language with the ranking in (5b), given in Tableau 2. Since TEL is now ranked higher than LE(CP), violation of the former is fatal, so that deletion of the complementizer becomes obligatory.

Tableau 2: ***Obligatory C-deletion in embedded declarative clauses***

	TEL	LE(CP)
a. .... [complementizer....]	*!	
b. □ .... [~~complementizer~~....]		*

Tableau 3 gives the evaluation according to the ranking in (5c), where the two constraints are in a tie, which is indicated in the tableau by means of a dashed line. Under this ranking the two rankings in (5a&b) are simultaneously active. Therefore, we have to read the tie in both directions: when we read the tie from left to right, the violation of LE(CP) is fatal (which is indicated by *>), and the (a)-candidate is optimal; when we read the tableau from right to left, the violation of TEL is fatal (which is indicated by *<), and the (b)-candidate is optimal. This predicts that deletion of the complementizer is optional in this case, since a candidate is only excluded by tied constraints when there is a fatal violation in both directions; cf. the discussion of Tableau 5.

Tableau 3: ***Optional C-deletion in embedded declarative clauses***

	LE(CP)	TEL

a. □ .... [complementizer.....]	*<
b. □ .... [~~complementizer~~.....]	*>

Note in passing that the ranking in (5c) accounts for the fact that the two constructions are both possible but has nothing to say about their relative frequency. This is not surprising given that we are dealing here with core syntax/competence; the relative frequency of the two constructions should rather be accounted for by some performance theory (which may be given shape as some version of stochastic OT, mentioned in footnote 4).

Let us now return to the difference between English and French with respect to the pronunciation patterns of relative clauses. It is clear that English is a language of type (5c) given that the complementizer is normally optional in embedded declarative clauses. French, on the other hand, is a language of type (5a): the complementizer is obligatory in embedded declarative clauses. Pesetsky has shown that this also accounts for the differences between the primeless English and French examples of (2) and (3), in which a bare relative pronoun is preposed, provided that we assume that in both languages the constraint RECOVERABILITY outranks the constraints TEL and LE(CP).

(6) a. French: REC >> LE(CP) >> TEL

b. English: REC >> TEL <> LE(CP)

Tableau 4 provides the evaluation of the primeless French examples in (3). Since the relative pronoun has a local antecedent it is recoverable after deletion, so that all candidates satisfy REC. The (b)-candidate is the optimal candidate because it is the only one that does not violate LE(CP); the fact that this candidate violates the lower-ranked constraint TEL is tolerated since this enables the satisfaction of the higher-ranked constraint LE(CP); cf. (1a).

Tableau 4: ***Relative clauses with preposed relative pronoun***

French		REC	LE(CP)	TEL
a.	l'homme [qui$_i$ ~~que~~ je connais t_i]		*!	
b. □	l'homme [~~qui~~$_i$ que je connais t_i]			*
c.	l'homme [~~qui~~$_i$ ~~que~~ je connais t_i]		*!	
d.	l'homme [qui$_i$ que je connais t_i]		*!	*

The evaluation of the corresponding English examples is given in Tableau 5, which is slightly more complex due to the fact that LE(CP) and TEL are in a tie. We are therefore dealing with two rankings at the same time: REC >> LE(CP) >> TEL and REC >> TEL >> LE(CP). The first ranking is the one we also find in French, and we have seen that this results in selection of the (b)-candidate as optimal. Under the second ranking, violation of TEL is fatal, so that the (a)- and the (c)-candidate are selected as optimal. As a result, three out of the four candidates are acceptable in English.

Tableau 5: ***Relative clauses with preposed relative pronoun***

English		REC	LE(CP)	TEL
a. □	the man [who$_i$ ~~that~~ I know t_i]		*>	

b. □	the man [~~who~~$_i$ that I know t_i]		*<
c. □	the man [~~who~~$_i$ ~~that~~ I know t_i]		*>
d.	the man [who$_i$ that I know t_i]		*> *<

Next consider the evaluation of the French examples in Tableau 6, in which a PP containing a relative pronoun is preposed. Since the preposition is not locally recoverable, deletion of it leads to a violation of the highest-ranked constraint REC: this excludes the (b)- and the (c)-candidate. Since the two remaining candidates both violate LE(CP), the lowest ranked constraint TEL gets the final say by excluding the (d)-candidate; cf. (1b). Note that this shows that the subranking LE(CP) >> TEL does not mean that the complementizer is always realized, but that this depends on the question whether it is preceded by some other element that must be realized; if so, TEL forces the complementizer to delete.

Tableau 6: ***Relative clauses with preposed PP***

French		REC	LE(CP)	TEL
a. □	l'homme [[avec qui]$_i$ ~~que~~ j'ai dansé t_i]		*	
b.	l'homme [[~~avec~~ qui]$_i$ que j'ai dansé t_i]	*!		*
c.	l'homme [[~~avec~~ qui]$_i$ ~~que~~ j'ai dansé t_i]	*!	*	
d.	l'homme [[avec qui]$_i$ que j'ai dansé t_i]		*	*!

Tableau 7 shows that we get the same result for the corresponding English examples: both the (b)- and the (c)-candidate are excluded by REC, and the (d)-candidate is excluded because it is harmonically bound by the (a)-candidate, that is, it has a fatal violation of TEL irrespective of whether we read the tie from left to right or from right to left. We will simply indicate violations of tied constraints that are fatal on all rankings available in the language by means of an exclamation mark.

Tableau 7: ***Relative clauses with preposed PP***

English		REC	LE(CP)	TEL
a. □	the book [[about which]$_i$ ~~that~~ he spoke t_i]		*	
b.	the book [[~~about~~ which]$_i$ that he spoke t_i]	*!		*
c.	the book [[~~about~~ which]$_i$ ~~that~~ he spoke t_i]	*!	*	
d.	the book [[about which]$_i$ that he spoke t_i]		*	*!

2.2.1 OT-syntax is a meta-theory or program, not a theory

Although OT-syntacticians agree that the language system consists of a generator that produces candidate sets and an evaluator that selects candidates from these sets as optimal in a given language L by means of the procedure in (1), they may have widely varying ideas on the nature of the generator and, as a result, the constraints that constitute the evaluator. The generator can take the form of virtually any imaginable generative device, as is clear from the fact that the current OT-approaches to syntax are based on very different and often incompatible linguistic theories. Some more or less random examples are given in (7).

(7) a. Lexical-Functional Grammar: Bresnan (2000); Sells (2001)
b. Early Principles-and-Parameters Theory: Grimshaw (1997); Pesetsky (1998)

c. Minimalism: Dekkers (1999); Broekhuis & Dekkers (2000); Heck & Müller (2000); Woolford (2007); Broekhuis (2008)
d. Others: Müller (2000/2001); Vogel (2006a)

Since the generators postulated by the proposals in (7) differ considerably and the generated candidate sets will therefore be constituted by candidates with entirely different properties, the postulated constraints will be quite different as well. This can be illustrated by comparing the OT-approaches proposed in Grimshaw (1997), Broekhuis (2008), and Dekkers (1999), which are all based on some version of the *principles-and-parameters* theory. Grimshaw's (1997) proposal was originally written in the early 90's and is based on the pre-minimalist *principles-and-parameters* framework. Among other things, this is clear from the fact that she tries to capture the directionality parameter, which was still generally adopted at that time, by means of two conflicting constraints HEAD LEFT and HEAD RIGHT (the head is leftmost/rightmost in its projection). In addition, she assumes the alignment constraints SPECIFIER LEFT and SPECIFIER RIGHT (the specifier is leftmost/rightmost in its projection). Given that Grimshaw also assumes that the structures created by the generator conform to the general X-bar-schema, the linearization of these structures follows from the language-specific ranking of these four constraints. Broekhuis (2008), which is based on the minimalist machinery proposed in Chomsky (2000) and later work, need not make use of Grimshaw's alignment constraints given that he adopts some form of the universal base hypothesis, according to which linear order is derived from the hierarchical relations between the constituents in the output representation (as expressed by Kayne's, 1994, Linear Correspondence Axiom). In his approach, linear order therefore follows from the language- specific ranking of a set of the so-called EPP-constraints, which favor movement of a goal into its probe's minimal domain, and the economy constraint *MOVE, which disfavors movement. For example, the "strong" ranking EPP(case) >> *MOVE requires movement of the probed noun phrase into the minimal domain of the unvalued case features of the verb or the inflectional node I, whereas the "weak" ranking *MOVE >> EPP(case) requires that the probe remain in its original position; see Section 3 for details. This proposal, which expresses the same intuition as Chomsky's Agree-based approach that Agree is normally sufficient for convergence, will find no place in OT-approaches that follow Groat & O'Neil (1996) in assuming that feature checking invariably triggers movement and that the linear order depends on the question whether it is the tail or the head of the resulting chain that is spelled out; such approaches will replace the EPP-constraints by, e.g., Dekkers' (1999) PARSE-F constraints, which favor pronunciation of moved constituents in the position of their formal feature (the head of the chain), and reinterpret *MOVE as a constraint that favors pronunciation of moved elements in their base position (the tail of the chain). This brief discussion demonstrates that properties of the proposed generator are immediately reflected in the nature of the postulated violable constraints of the OT-evaluator. The differences between the three OT-approaches discussed here are relatively small due to the fact that the proposed generators all find their origin in the Chomskyan generative tradition, but it will be clear that the differences between these OT-approaches and OT-approaches that are based on other traditions may be much larger. Note that the choice of the correct generator and the selection of the correct set universal constraints are, of course, both empirical issues.

Combining the Minimalist Program and Optimality Theory

When we compare Figure 1 and Figure 2, we see immediately that the overall structure of Chomsky's version of MP has much in common with the model assumed in standard OT. Both

have a generative device that defines a set of possible structures, from which languages select a subset of acceptable sentences by means of some filtering device. From the discussion in the preceding sections, it will have become clear that the two devices are not equally well developed in the two programs. Minimalist research has focused mainly on the generative device, despite the fact that the following quote from Chomsky (1995:220) shows that a filtering device was postulated right from the start:

"The language L thus generates three relevant sets of derivations: the set D of derivations, a subset D_C of convergent derivations of D, and a subset D_A of admissible derivations of D. [Full Interpretation] determines D_C, and the economy conditions select D_A. [...] D_A is a subset of D_C".

Although the filtering device has been endowed with various names in the respective stages of the minimalist framework (they have been referred to as global economy, bare output, interface and effect-on-output conditions), relatively little work has been devoted to developing a coherent theory of it. The situation in OT is rather the reverse: much work has been devoted to the substantive content of the filtering device (that is, the constraints and their ranking), but virtually no attention has been paid to the generator. Given this situation it might be useful to combine the two approaches by assuming that the generative device is some version of the computational system C_{HL}, and that the filtering device is some version of the OT-evaluator, as in Figure 3.

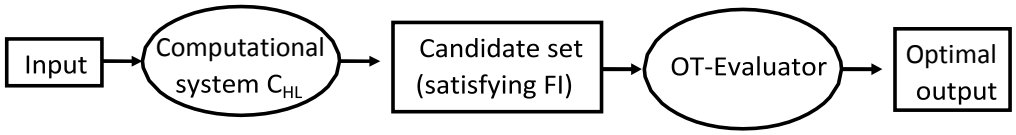

Figure 3: The architecture of grammar

A potential advantage of this hybrid MP + OT model is that it provides OT-syntax with an explicitly formulated generator and MP with at least a general format for expressing the interface conditions. Furthermore, now that both devices have been assigned an explicit format, we can seriously investigate the division of labor between the two components. For example, Broekhuis and Dekkers (2000) have noted that Pesetsky's RECOVERABILITY is rather suspect as an OT-constraint given that it is never violated: it is always the *highest* ranked constraint. This suggests that we are actually dealing with an inviolable condition on the operation Delete, which must therefore be added to the inventory of syntactic operations in C_{HL}. Given that the derivation is cyclic, the postulation of Delete makes it impossible to account for the recoverability restriction by appealing to the availability of some local antecedent. Therefore, the restriction must rather be formulated in terms of semantic features, which will have various ramifications for the analysis of the pronunciation patterns of relative clauses. We will not digress here on this specific issue any further but refer the reader to Dekkers (1999) for further discussion.

An important point is that the hybrid model may substantially change our views on the properties ascribed to the generative and filtering device by traditional MP/OT. Our hope is that we can simplify current MP by expressing all language-specific statements (including the more recent effect-on-output conditions discussed above) by means of language-specific rankings of otherwise universal violable constraints. For example, MP stipulates that languages differ with respect to question of whether functional heads force movement of the phrases with which they are in an Agree-relation. A formal way of expressing this is by assuming that such heads may or

may not have an epp-feature. The following section will show that it is not only readily possible to replace the notion of epp-feature by a small set of violable constraints, but that doing this also results in a descriptively more adequate theory.

Conclusion

Adding an OT-evaluation to MP can be done in a very 'minimal' way, simply by introducing the idea that the output of the computational system is filtered in an optimality-theoretic fashion by means of a language-specific ranking of otherwise universal constraints. The introduction of such an optimality-theoretic evaluator eliminates the need for many other devices that are currently used in MP to capture cross-linguistic differences such as language-specific filters of the type in (16) and parameter settings.

Recommendations

Every writer should ensure that in their write-up Minimalism rules should be applied where by elimination of excessive complexity of principles is observed. This can be done by creating a model of language that eliminates unnecessary stops in the representation of the derivation of a sentence. There is need, in every write-up, to minimize the theoretical constructs, structure and operations.

Godfrey K. DANIEL, *Ph.D*

REFERENCES

Archangeli, Diana. (1997). *Optimality Theory*. An introduction to linguistics in the 1990s. In Optimality theory: an overview, eds. Diana Archangeli and Terence Langendoen, 134-170. Malden/Oxford, Blackwell.

Bresnan, Joan. (2000). *Optimal Syntax*. In Optimality Theory: phonology, syntax and acquisition, eds. Joost Dekkers, Frank Van der Leeuw and Jeroen Van de Weijer, 334-385. Oxford, Oxford University Press.

Broekhuis, Hans. (2008). *Derivations and evaluations: object shift in the Germanic languages*. Berlin/New York: Mouton de Gruyter.

Broekhuis, Hans, and Dekkers. Joost (2000). The minimalist program and optimality theory: derivations and evaluations. In *Optimality Theory: phonology, syntax and acquisition*, eds. Joost Dekkers, Frank van der Leeuw and Jeroen van de Weijer, 386-422. Oxford/New York, Oxford University Press.

Chomsky, Noam, and Howard Lasnik. (1977). Filters and control. *Linguistic Inquiry* 8:425-504.

Chomsky, Noam. (1995). *The minimalist program*. Cambridge (Mass.): MIT Press.

Chomsky, Noam. 2000. Minimalist inquiries: the framework. In *Step by step. Essays on minimalist syntax in honor of Howard Lasnik*, eds. Roger Martin, David Michaels and Juan Uriagereka, 89-155. Cambridge (Mass.), MIT Press.

Chomsky, Noam. 2001. Derivation by phase. In *Ken Hale. A life in Language*, ed. Michael Kenstowicz, 1-52. Cambridge (Mass.), MIT Press.

Dekkers, Joost. 1999. Derivations & Evaluations. On the syntax of subjects and complementizers, University of Amsterdam/HIL: PhD. thesis.

Grimshaw, Jane. 1997. Projection, heads and optimality. *Linguistic Inquiry* 28:373-422.

Groat, Erich, and John O'Neil. 1996. Spell-Out at the LF interface. In *Minimal Ideas. Syntactic studies in the minimalist framework*, eds. Werner Abraham, Samuel David Epstein and Höskuldur Thráinsson, 113-139. Amsterdam/Philadelphia, John Benjamins.

Heck, Fabian, and Gereon Müller. 2000. Successive Cyclicity, Long-Distance Superiority, and Local Optimization. In *WCCFL 19*, eds. Roger Billerey and Brook Danielle Lillehaugen, 218-231. Somerville (Mass.) Cascadilla Press.

Kayne, Richard S. 1994. *The antisymmetry of syntax*. Cambridge (Mass.): MIT Press

Müller, Gereon. 1999. Optionality in Optimality-Theoretic Syntax. *Glot Interntional* 4:5:3-8. Müller, Gereon. 2000. Shape conservation and remnant movement. In *Proceedings of NELS 30*, eds. A. Hirotani, N. Hall Coetzee and J.-Y. Kim, 525-539. Amherst (Mass.), GLSA.

Müller, Gereon. 2001. Order preservation, parallel movement, and the emergence of the

unmarked. In *Optimality-theoretic syntax*, eds. Géraldine Legendre, Jane Grimshaw and Sten Vikner, 113-142. Cambridge (Mass.)/London, MIT Press/MITWPL.Pesetsky, David. 1997. Optimality theory and syntax: movement and pronunciation. In *O*

ptimality theory, eds. Diana Archangeli and Terence Langendoen. Malden/Oxford, Blackwell.

Pesetsky, David. 1998. Some optimality principles of sentence pronunciation. In *Is the best good enough?* eds. Pilar Barbosa, Danny Fox, Paul Hagstrom, Martha McGinnis and David Pesetsky, 337-383. Cambridge (Mass.)/London, MIT Press/MITWPL

Sells, Peter. 2001. *Structure Alignment and optimality in Swedish*. Stanford: CSLI Publications.

Vogel, Ralf. 2006a. The simple generator. In *Optimality Theory and Minimalism: A Possible Convergence? Linguistics in Potsdam 25*, eds. Hans Broekhuis and Ralf Vogel. University of Potsdam: http://www.ling.uni-potsdam.de/lip.

Woolford, Ellen. 2007. Case locality: Pure domains and object shift. *Lingua* 117:1591-1616.