Crash-proof syntax and filters

Hans Broekhuis (Leiden U.) and Ralf Vogel (U. of Bielefeld)

Revised Version 1.1: February 18, 2009

1 Introduction

The crash-proof syntax project aims at developing a derivational system that is optimal in the sense that it "generates only objects that are well-formed and satisfy conditions imposed by the interface systems". The target system should further have the property that "no filters are imposed on the end products of derivations, and no global filters (e.g. comparison of derivations) assign status to derivations as a whole"; cf. Frampton & Gutmann (2002:90). In this contribution, we are supposed to provide the outlook of the OT-community on this project, but we want to say immediately that performing this task successfully is impossible due to the fact that the OT-approaches to syntax are simply too diverse to make any statement of sufficient generality. Of one thing we are sure, though, and that is that syntacticians working within the various OT-approaches will disagree with Frampton & Gutmann's claim that (transderivational) filters on the output of the computational system should be abandoned; cf. Section 2. For this reason, Section 3 will focus on this aspect of the proposal; we will argue that even when it turns out to be possible to develop a crash-proof syntax that only generates well-formed objects that satisfy the interface conditions, filters on the output of the computational system will remain an essential ingredient of the theory of syntax. Section 4 will show that this does not necessarily imply, however, that the more general and modest aim of the crash-proof syntax project to limit the output of the derivational system to "objects that are well-formed and satisfy conditions imposed by the interface systems" should be dismissed as irrelevant, and that for certain types of OT-syntax developing a generative device that meets that requirement may be highly desirable.

2 OT-syntax as a theory of filters

It has been stressed over and over again that minimalism is not a theory but a program. To quote Boeckx (this volume), the minimalist program "refers to a family of approaches that share a few basic assumptions and guiding intuitions concerning the nature of the language faculty, cognition, and biology, but otherwise differ significantly from one another in how these guidelines are articulated". Boeckx' description of the notion of program also applies to optimality theory (OT). OT refers to a family of linguistic theories that adopts the general linguistic model in Figure 1. 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. Furthermore, OT adopts the basic assumption that the evaluator consists of violable constraints, and that it is the language-specific ranking of these constraints that determines which candidates from the Candidate Sets are optimal in language L. We will also show that the existing OT-approaches differ significantly from one another in that the generators postulated by them can differ in a multitude of ways: consequently, the generated Candidate Sets may differ considerably among the different OT-approaches with the concomitant result that the set of postulated OT-constraints may also vary wildly.

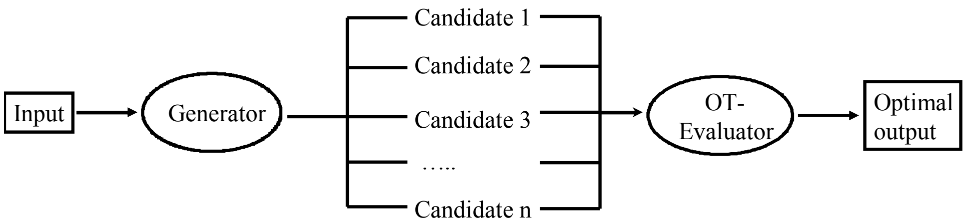

Figure 1: Optimality-theory

It must be stressed that the guiding intuition of OT expressed by Figure 1 is in fact not much different from the one that has been assumed in mainstream generative grammar from the mid seventies until now, and which was first made explicit in Chomsky & Lasnik (1977). When we put aside the fact that it was generally assumed until the change of the millennium that the derivation of the LF- and PF-representation split at some point in the derivation (S-Structure or Spell-out point), the main difference concerns the *form* of the evaluator: in mainstream generative syntax, the evaluator is taken to consist of universal principles and language-specific filters, whereas OT claims that such principles and filters can be more adequately expressed by means of the ranking of more primitive violable constraints; see Pesetsky (1997; 1998) and Dekkers (1999) for early demonstrations of this, and Broekhuis (2006; 2008) for a more general discussion.

OT can furthermore be called a program as there are a few basic assumptions about grammar and language that are truly genuine for OT, at least within generative grammar, and which unite the otherwise very divergent OT-literature. One of these is a holistic conception of language in the sense that the grammaticality of an expression E for some language L cannot be established by inspecting E alone, but is determined by comparison to other expressions produced by the generator in a way that normally goes far beyond what is discussed under the term transderivationality in early minimalism. For instance, a derivation that is blocked by a transderivational constraint yields an ungrammatical expression in minimalism. A loser in one OT-competition, on the other hand, might nevertheless be the winner of another competition; the Icelandic object shift examples in (3) in Section 3 will provide a concrete example of this.

The model in Figure 1 entails two notions of well-formedness: one with respect to the generator and the other with respect to the evaluator. The former can be seen as the OT-counterpart of the early minimalist notion of possible (but possibly non-convergent) derivation, whereas the latter one is the OT-instantiation of grammaticality. The usual version of the generator in OT is rather liberal and unrestricted, and allows for a comparatively large candidate set; each member of this candidate set is well-formed according to the rules of the postulated generator, but not necessarily grammatical. Since generator-related well-formedness is not determined holistically, it can be conceived of in a minimalist style, which is in fact quite common. It might even be possible and desirable to make the OT-generator crash-proof, given that it can be formulated independently of any considerations about grammaticality or acceptability; see Heck & Müller (2000) for an early attempt within an OT-version of phase theory, which seems pretty similar in spirit to what Frampton & Gutmann propose. The evaluator of OT-syntax, on the other hand, often incorporates aspects of the interpretative systems; Vogel (2004, 2006a) and Broekhuis (2008), for instance, have developed an explicit formulation of the OT-evaluator that contains interface constraints. If these observations are correct, then the quest for the best version of crash-proof syntax in minimalism and the quest for the best formulation of the OT-syntax generator may have very much in common. We will return to this issue in Section 4.

In a sense OT can also be seen as a meta-theory or a methodological guideline, which is clear from the fact that, contrary to minimalism, it may be applied to a wide variety of empirical domains: it can be equally well applied to phonology as to, e.g., syntax, and it is certainly conceivable that it can be successfully applied outside the domain of linguistics as well. However, when we restrict ourselves to a certain empirical domain, it may be the case that the differences between the different OT-approaches are so small that it is actually justified to speak of a more or less coherent theory. This might well be the case for OT-phonology, given that there seems to be considerable agreement among OT-phonologists on the nature of the input, the operations that can be performed by the generator, and the nature of the output. Furthermore, OT-phonologists do not only agree on the basic assumption that the evaluator consists of ranked violable constraints, but they also seem to share the belief that the postulated constraints are of just two types, the so-called faithfulness and markedness constraints. And, finally, there even seems be some consensus about the individual constraints that are needed. Of course, there are also hotly debated issues, such as the question of whether the constraints are part of an innate, universally available set CON, or whether they are acquired on the basis of the primary linguistic data.

The situation in OT-syntax is entirely different: we are clearly not dealing with a generally accepted theory. We already noted that the model in Figure 1 is very specific about the nature of the evaluator, which has the defining property of consisting of ranked violable constraints, but the nature of the generator is left open entirely; the generator can take the form of virtually any imaginable generative device, and, as a result, the generators of the current OT-approaches to syntax are based on different and often incompatible linguistic theories. Some more or less random examples are given in (1).

(1)
a. Lexical-Functional Grammar: Bresnan (2000); Sells (2001)
b. Early Principles-and-Parameters Theory: Grimshaw (1997); Pesetsky (1998)
c. Minimalism: Dekkers (1999); Woolford (2007); Broekhuis (2008)
d. Others: Müller (2000/2001); Vogel (2006a)

Since the generators postulated by the proposals in (1) 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. As a result, we are dealing with OT-approaches that are as different as (or perhaps even more different than) the theories on which the generator is modeled. We will illustrate this below 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 assumed 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 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 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 version of Kayne's (1994) Linear Correspondences Axiom, according to which linear order is derived from the hierarchical relation between the constituent in the output representation. 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 (in the sense of Chomsky

1995: ch.3), and the economy constraint *MOVE, which disfavors movement. For example, the “strong” ranking EPP(φ) >> *MOVE requires movement of the probed noun phrase into the minimal domain of the unvalued φ-features of the verb or the inflectional node I, whereas the “weak” ranking *MOVE >> EPP(φ) requires that the probe remain in its original position. The EPP-constraints, which are used to express 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., Dekker’s (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).

The previous paragraph has shown 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 above are still 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 (generative) traditions may be much larger. For example, Broekhuis (2008) and Sells (2001) both develop an OT-analysis of Scandinavian object shift, but the two proposals differ at least as much as the minimalist and Lexical-Functional approaches that they are based on: whereas Broekhuis’ analysis is built on the restrictions on movement of the clausal constituents, Sells’ analysis is based on the restrictions on their phonological alignment. To drive things to the extreme, it is even imaginable to postulate a universal evaluator, but a language particular generator; from an OT-perspective, this would still count as a version of Universal Grammar.

The observations given in this section make it hard to formulate a generally valid assessment of the crash-proof syntax project from an OT-perspective. For example, OT-approaches that do not postulate movement or formal features with the properties attributed to them in the minimalist program will simply have nothing to say on the issue. This is different, however, for approaches like the ones developed in Dekkers (1999) or Broekhuis (2008), which assume that the generator in Figure 1 is some version of the computational system postulated in the minimalist program.

3 Crash-proof syntax does not void the need for filters

This section will focus on Frampton & Gutmann’s claim that crash-proof syntax should have the property that “no filters are imposed on the end products of derivations, and no global filters (e.g. comparison of derivations) assign status to derivations as a whole”. Since we have seen in the previous section that OT-syntax simply provides a more sophisticated theory of language-specific filters, this claim amounts to saying that there can be no such thing as a minimalist OT-approach of the sort developed in, e.g., Dekkers (1999) or Broekhuis (2008). We will argue in this section that such a claim is not tenable, given that the elimination of filters can only be obtained with the considerable cost of rejecting at least one of the basic assumptions/guiding intuitions of the minimalist program, namely that the computational system C_{HL} is invariant in the sense that it does not exhibit language-specific properties. In other words: eliminating the filtering device will place crash-proof syntax outside the minimalist tradition as we currently understand it.

3.1 Movement

In Section 2, we have shown that the OT-model in Figure 1 is compatible with virtually any imaginable syntactic approach that postulates a generative device. The sole exception would be a theory with a generative device that does not overgenerate, because in that case the OT-evaluator would have nothing to do, and hence be superfluous. This means that in order to eliminate the OT-evaluator, crash-proof syntax must meet the *all-and-only* requirement that was posed on earlier transformational-generative grammar: for each language L, the generative device must generate all representations that are acceptable in L, and no others. We will show in this subsection that this condition is incompatible with the minimalist basic assumption/guiding intuition that the computational system C_{HL} is invariant.[1]

Postulating an invariant C_{HL} would be compatible with Frampton & Gutmann's goal of eliminating the OT-evaluator when *all* representations generated by crash-proof syntax are well-formed in *all* languages. This is, however, obviously not a property of the Agree-based system that they adopt. Consider the simple structure in (2), and assume that the unvalued feature on X can be valued by the feature on Y, as in (2b). Given that X and Y agree, the later can be moved into the minimal domain of the former, as in (2c).

(2) a. $X_{[uF]}$ .... $Y_{[vF]}$
b. $X_{[vF]}$ .... $Y_{[vF]}$
c. [$Y_{[vF]}$ [$X_{[vF]}$ ... ~~Y~~]]

Both (2b) and (2c) satisfy the interface conditions and should therefore be derived by crash-proof syntax. It is, however, not the case that both representations will be allowed in all languages: languages typically differ in word order, which is taken to imply that languages differ with respect to the movements they do or do not allow, provided at least that one assumes some version of Kayne's (1994) Linear Correspondences Axiom. Consequently, we need some language-specific means that may force or block the movement in (2c). There are at least two ways of doing this, while adhering to the minimalist basic assumption/guiding intuition that the computational system C_{HL} does not exhibit language-specific properties.

The first, traditional minimalist way of making the distinction between languages that do or do not have the movement in (2c) is to assume that languages differ in the presence or absence of an additional lexical property on probe X that forces movement of its goal Y, the so-called strength or epp-feature: when X has this additional feature, the movement in (2c) must apply; when X lacks this feature, the movement in (2c) is blocked for reasons of economy. The second way of making the distinction is to postulate some language-specific filter that selects the resulting representation with or without movement as the optimal one. As we have already seen in Section 2, this can readily be expressed in an optimality-theoretic manner by the interaction of the constraint EPP(F), which favors movement of goal Y into the minimal domain of probe X, and the economy constraint *MOVE, which disfavors this movement. A third option would be to attribute some language-specific property to the

---

[1] We believe that derivational theories of the sort under discussion should adopt as their null hypothesis that the syntactic representations fully determine the word order of the resulting utterances, and thereby reject the current minimalist tendency to postulate stylistic movement rules in the PF-component of the grammar in order to overcome shortcomings in the design of C_{HL}; only in this way can we avoid the risk that we are simply sweeping dust under the carpet. Furthermore, to our knowledge, Holmberg's (1999) theory of object shift is the only example of a derivational theory with phonological movement that has been worked out in sufficient detail to be tested, and it is clear that this theory is incompatible with the generally accepted claim that phonological movement cannot affect semantic interpretation; cf. Chomsky (2001:32ff.). A number of other serious (conceptual) problems for Holmberg's proposal can be found in Vogel (2006b) and Broekhuis (2008:§4.3.3).

computational system C_{HL} itself, but this should be dismissed immediately given that this would go against the minimalist basic assumption/guiding intuition that C_{HL} is invariant.

The traditional minimalist approach to the problem does not seem tenable, given that it gives rise to a too rigid, *all-or-nothing* system: when X is marked with an epp-feature in the lexicon, we predict that the movement in (2c) will always apply, and when X is not marked with an epp-feature, we predict that this movement will always be excluded. Thus, we wrongly exclude languages with movements that only apply under certain conditions. A well-known example of such movements is Icelandic object shift in (3a), which is pragmatically conditioned in the sense that it can only apply when the object is part of the presupposition (old information) of the clause; when the object belongs to the focus (new information) of the clause object shift is blocked.

(3) a. Jón keypti bókina$_i$ ekki t_i — *bókina* ⊂ presupposition
Jón bought the book not
b. Jón keypti ekki bókina — *bókina* ⊂ focus

This problem has led Chomsky (2001) to assume that having an epp-feature is not an inherent property of probes, but that epp-features can be freely added provided that they will have some semantic effect on the output: when the object remains VP-internally it will be interpreted as the focus of the clause (provided, at least, that it is situated at the phonological border of the VP), whereas it will be interpreted as part of the presupposition of the clause when it is placed in the minimal domain of its probe *v*. Given that pragmatically conditioned object shift is not universally available (in languages that meet Chomsky's additional condition that the main verb evacuates the VP in simple past/present tense constructions; cf. French), the postulation of this *language-specific* Effect-on-Output condition on object shift is a radical break with Chomsky's (1995) earlier minimalist credo that the sole locus of language variation is the lexicon, and means that language-specific output filters are introduced to account for this type of variation.

In fact, the introduction of the Effect-on-Output condition makes the postulation of epp-features entirely superfluous: it is sufficient to assume that (3a) and (3b) are part of the same Candidate Set and to formulate a filter that requires that the object be moved under the appropriate condition. In Broekhuis (2008), for example, the contrast in (3) is accounted for by assuming that Icelandic has a "strong" ranking EPP(case) >> *MOVE, which predicts that object shift is "normally" obligatory, but that there is an additional constraint ALIGNFOCUS, which was first proposed by Costa (1998), which outranks EPP(case) in Icelandic and disfavors object shift by requiring that new information be aligned with the right edge of the clause. The OT-evaluations in the following tableaux show that this accounts for the fact that object shift is obligatory in (3a), but blocked in (3b).

Tableau 1: Icelandic

object ⊂ presupposition	ALIGNFOCUS	EPP(case)	*MOVE
Jón keypti ekki bókina		*!	
Jón keypti bókina$_i$ ekki t_i ☞			*

Tableau 2: Icelandic

object ⊂ focus	ALIGNFOCUS	EPP(case)	*MOVE
Jón keypti ekki bókina ☞		*	
Jón keypti bókina$_i$ ekki t_i	*!		*

Note that the advantage of replacing Chomsky's Effect-on-Output condition by an OT-evaluation of this sort is that the postulated constraints are more primitive and can be used in a wider range of cases. This can be illustrated by means of so-called A-Scrambling in Dutch.[2] Since A-Scrambling has the hallmark of Icelandic object shift of being sensitive to the information structure of the clause, Broekhuis (2008) has claimed that these languages have the same ranking of the three constraints proposed above. In addition, however, A-Scrambling is sensitive to scope, as shown by the examples in (4a&b), which differ with respect to the relative scope of the quantified adverb *vaak* 'often' and the universally quantified object *alle boeken* 'all books'. Example (4a) expresses that Jan often takes along the full set of books; the contention is false if the cardinality of the contextually determined set of books is ten and Jan only occasionally takes more than nine books with him. Example (4b), on the other hand, expresses that it holds for each of the books individually that they are often taken along by Jan; the contention can be true in the given context when Jan always takes less than ten books with him; cf. Broekhuis (2008:222).

(4) a. dat Jan vaak alle boeken meeneemt. (vaak > ∀)
that Jan often all books takes-along
b. dat Jan alle boeken$_i$ vaak t_i meeneemt. (∀ > vaak)

These examples can be readily accounted for by assuming the constraint SCOPE, which favors that the relative scope of an argument corresponds to its topmost A-position. The Dutch data will follow when we assume that SCOPE outranks EPP(case).

Tableau 3: Dutch

Scope: vaak > ∀	SCOPE	EPP(case)	*MOVE
dat Jan vaak alle boeken meeneemt ☞		*	
dat Jan alle boeken$_i$ vaak t_i meeneemt	*!		*

Tableau 4: Dutch

Scope: ∀ > vaak	SCOPE	EPP(case)	*MOVE
dat Jan vaak alle boeken meeneem	*!	*	
dat Jan alle boeken$_i$ vaak t_i meeneemt ☞			*

The Icelandic and Dutch examples in (3) and (4) illustrate nicely how the more atomic nature of the violable OT-constraints does not only make the system more flexible than the language-specific filters postulated by the traditional approach, but also makes the system more transparent by showing what additional conditions are imposed by the performance systems on the optimal candidate. This is very clear in the case of ALIGNFOCUS and SCOPE, which are simply formalizations of earlier observations that, in some languages, word order is sensitive to the information structure of the clause and/or relative quantifier scope. This enhanced transparency is very welcome given that the conditions on object shift are much more complex than suggested by the discussion so far. Icelandic object shift, for example, is also subject to Holmberg's generalization, that is, may not cross the main verb in VP-internal position; as a result, the object cannot be moved in complex verb constructions like (5), and

[2] Next to A-Scrambling Dutch has A′-Scrambling, which is not restricted to DPs and is related to notions like contrastive/exhaustive focus and negation; see Section 3.2 for a brief discussion of the latter case. We will ignore A′-movement here and refer to Neeleman (1994), Broekhuis (2008:§2.1), and the references cited there for more extensive discussion.

sentence (5a) is consequently ambiguous between the focus and the presupposition reading for the object; cf. Holmberg (1986), Vikner (2001), and many others.

(5) a. Jón hefur ekki [$_{VP}$ keypt bókina]. *bókina* ⊂ focus or presupposition
Jón has not bought the book
b. *Jón hefur bókina ekki [$_{VP}$ keypt t_{bokina}]

Chomsky (2001) realizes this and builds this into his filter for Icelandic object shift, which as a result becomes rather complex. This complexity might be a problem from the perspective of language acquisition, given that we are again dealing with a language-specific property; Dutch A-scrambling, for example, does not exhibit this restriction. The difference between Icelandic and Dutch can, however, again be readily and transparently accounted for by postulating a constraint that blocks movement of an object across the verb. Broekhuis (2008:ch.3), for example, claims that this is the result of the order-preservation constraint HEAD-COMPLEMENT, which favors preservation of the underlying order of the verb and the direct object: in Icelandic, this constraint outranks EPP(CASE) with the result that object shift will be blocked in the complex verb constructions in (5), irrespective of whether the object is part of the presupposition or the focus of the clause; cf. the tableaux below. In Dutch the constraint HEAD-COMPLEMENT is ranked lower than EPP(CASE), and the object will be able to freely cross the main verb in order to satisfy this constraint, but this will go unillustrated here.[3] Note that the dotted line in the Tableaux below simply indicates that it is not possible to determine the relative order of the constraints ALIGNFOCUS and HEAD-COMPLEMENT on the basis of the Icelandic data given here: both orders of these constraints will give rise to the desired result.

Tableau 5: Icelandic

object ⊂ presupposition	ALIGNFOCUS	HEAD-COMPLEMENT	EPP (case)	*MOVE
Jón hefur ekki [$_{VP}$ keypt bókina] ☞			*	
Jón hefur bókina ekki [$_{VP}$ keypt t_i]		*!		*

Tableau 6: Icelandic

object ⊂ focus	ALIGNFOCUS	HEAD-COMPLEMENT	EPP (case)	*MOVE
Jón hefur ekki [$_{VP}$ keypt bókina] ☞			*	
Jón hefur bókina ekki [$_{VP}$ keypt t_i]	*!	*!		*

We can readily add to this complexity: Scandinavian languages like Danish differ from Icelandic in that they do not exhibit object shift of full noun phrases, although they do have pronominal object shift, which is again sensitive to the position of the verb; cf. Vogel (2006b) and Broekhuis (2008:ch.3) for detailed discussion. This can readily be accounted for by postulating a constraint that favors movement of unstressed pronouns. Furthermore, all languages mentioned so far require that the order of the arguments remain unaffected by

[3] The discussion of Dutch is simplified here: it is actually not the ranking of EPP(CASE) and *MOVE that is relevant, but the ranking of EPP(φ) and *MOVE that is relevant; see Broekhuis (2008: ch.3) for a more careful discussion. Note further that, strictly speaking, the appeal to linear order in the main text cannot be correct given that this is not a syntactic notion: order-preservation constraints like HEAD-COMPLEMENT should therefore be phrased in purely hierarchical terms, which is of course readily possible when we adopt some version of Kayne's Linear Correspondences Axiom.

object shift, which can be readily accounted for by assuming some highly ranked order-preservation constraint that disfavors changing the underlying order of the arguments (cf. Müller, 2000/2001, and Broekhuis, 2008:ch.3, for detailed discussion); German differs from all these languages in that A-scrambling may change the order of the arguments, and thus has a low ranking of this constraint. It will be clear that it will not be very enlightening (or easy) to build all these complexities into the language-specific filters on object shift; especially from the perspective of language acquisition the OT-approach seems much favored given that the child will find ample evidence in the primary linguistic data to order the relevant (universally available) constraints.

The discussion in this subsection has shown that even when we adopt some version of crash-proof syntax, it will remain necessary to filter the output of the generator. It must be noted here, of course, that Frampton & Gutmann (2006) suggest in their discussion of Icelandic object shift that this conclusion can be avoided by making the computational system itself sensitive to the interpretative effect of object shift, but we have already argued that this is only possible by dropping the minimalist basic assumption/guiding intuition that the computational system C_{HL} is invariant in the sense that it does not exhibit language-specific properties. In fact, Frampton & Gutmann's suggestion implies a second, even more radical break with the minimalist tradition, given that it is not compatible with the postulate of the autonomy of syntax: this section has shown that a full crash-proof account of Scandinavian object shift and Dutch/German A-scrambling requires that the computational system be made sensitive to a wide variety of non-syntactic properties of the resulting utterances. The Icelandic data in (3) show that the computational system must be made sensitive to the pragmatically defined notions focus and presupposition, the Dutch examples in (4) show that also the meaning of the output representations may play a role, and the fact that Danish object shift is restricted to weak pronouns strongly support Vogel's (2006b) claim that the phonological properties of the output representation are also relevant. A third problem is that all our objections against Chomsky's filter approach carry over to Frampton & Gutmann's version of crash-proof syntax: incorporating the language-specific restrictions on, e.g., object shift into the generator will give rise to a very complex version (or rather versions) of C_{HL}; it may actually be impossible given that earlier attempts to derive Holmberg's Generalization on object shift from the Minimal Link Condition have failed; see Broekhuis (2008:§3.1) for a comprehensive review of the relevant literature on this issue.

3.2 Negative sentences

This subsection provides another example that will show that the postulation of language-specific filters is unavoidable: the distribution of negative polarity items (NPIs) in English and Dutch. First, consider the English examples in (6): examples like (6a) show that in simple clauses sentential negation is normally expressed by means of the negative adverb *not*, which can be followed by a negative polarity item like *anybody*; examples like (6b), in which sentence negation is expressed by means of a negative noun phrase, are as a rule far less acceptable. More can be said on the acceptability of negative noun phrases in simple English sentences, but we refer the reader to Broekhuis & Klooster (2007) for a more careful discussion and take the contrast in (6) as the proper idealization.

(6) a. John is not satisfied with anybody.
b. *John is satisfied with nobody.

The examples in (7) show that Dutch exhibits the mirror image of English: sentential negation can be expressed by means of a negative noun phrase, whereas the use of the negative adverb *niet* followed by an NPI leads to ungrammaticality (irrespective of the placement of the NPI).

(7) a. Jan is over niemand tevreden.
Jan is about nobody satisfied
b. *Jan is niet tevreden over ook maar iemand.
Jan is not satisfied about anybody

For concreteness' sake, let us follow Haegeman (1995) in assuming that negative clauses contain a NegP, and that the head of the NegP must be checked/valued by some negative element, which must subsequently be placed in SpecNegP. The Dutch and English examples in (6) and (7) show that there are at least two possibilities to obtain this. The first option is selected by Dutch: the clause contains some negative phrase, which is probed by Neg and subsequently moved into SpecNegP; the movement is visible in (7a) due to the fact that the PP *over niemand* is a complement of the adjective *tevreden*, but cannot occupy the unmarked, post-adjectival position. The second option, which is strongly preferred in English, is to check/value Neg by merging the negative adverb *not* in SpecNegP, and realize the PP-complement as an NPI.

Broekhuis & Klooster (2007) have argued that the contrast between the English and Dutch examples in (6) and (7) poses two problems for the traditional minimalist approaches. The first concerns the generally adopted minimalist claim that the numeration/initial array fixes "the *reference set* for determining whether a derivation from [the initial array] A is optimal —that is, not blocked by a more economical derivation" (Chomsky 1995:227). This means that, by definition, derivations based on different numerations are not in competition and are therefore unable to block each other. However, this is exactly what is happening in the examples in (6) and (7); examples that contain a negative noun phrase have the numeration in (8a), whereas examples with a negative adverb and an NPI have the numeration in (8b), but nevertheless they seem to be able to block each other.

(8) a. {..., Neg, $NP_{[+neg]}$, ...}
b. {..., Neg, *niet*/*not*, NPI, ...}

Given Chomsky's (1995:§4.9) claim that examples like (9a) and (9b) are both acceptable due to the fact that they are based on different numerations, we would expect that the English examples in (6) or the Dutch examples in (7) could likewise co-exist. The fact that this expectation is not borne out seriously undermines the claim that the numeration/initial array determines the reference set; see Broekhuis & Klooster (2007) for more extensive discussion.[4]

(9) a. There seems to be someone in the room
b. Someone seems to be in the room

---

[4] Note that Frampton & Gutmann (1999/2002) also argue against the claim that derivations take a numeration as their starting point. Their account of the co-occurrence of the examples in (9) is based on the claim that indefinite noun phrases may be either an NP or a DP. Broekhuis & Klooster (2007) follow Moro (1997) in assuming that English expletive constructions are actually Locative Inversion constructions, which they motivate by showing that a similar contrast as in (10), which is to be discussed shortly, appears when Locative Inversion applies in raising contexts: since neither Chomsky's nor Frampton & Gutmann's proposal has anything to say about this similarity, Moro's approach can be considered superior.

The second problem concerns Chomsky's (1995:226) claim the operation Merge (= external merge) is inherently more economical than the operation Move (= internal merge) and therefore blocks the application of the latter. This claim plays an important role in his account of the contrast in (10): assuming that at each stage of the derivation we select the most economical option available, at the stage [to be someone here], external merge of the expletive is preferred to internal merge of *someone* for reasons of economy, and as a result (10b) cannot be derived.

(10) a. There$_i$ seems [$_{IP}$ *t*$_i$ to be someone in the room]
b. *There seems [$_{IP}$ someone$_i$ to be *t*$_i$ in the room]

Given that we have concluded that the a- and b-examples in (6) and (7) are in competition, the claim that Merge is inherently more economical than Move wrongly predicts that both English and Dutch would opt for the NPI-constructions. Broekhuis & Klooster conclude from this that Merge and Move are both costly operations, which they express by assuming the economy constraints in (11).

(11) a. *MOVE: Do not move
b. *MERGE: Do not merge

The contrast between English and Dutch can now be expressed by claiming that the two languages differ in the relative ranking of these constraints. In Dutch, *MERGE outranks *MOVE with the result that the movement option is preferred to the NPI option, whereas in English *MOVE outranks *MERGE with the result that the NPI option is preferred to the movement option.

Tableau 7: Dutch

Sentential negation	*MERGE	*MOVE
... [$_{NegP}$ niemand$_i$ [Neg ... *t*$_i$ ...]] ☞		*
... [$_{NegP}$ niet [Neg ... NPI ...]]	*!	

Tableau 8: English

Sentential negation	*MOVE	*MERGE
... [$_{NegP}$ nobody$_i$ [Neg ... *t*$_i$ ...]]	*!	
... [$_{NegP}$ not [Neg ... NPI ...]] ☞		*

The data discussed in this subsection are not only a problem for the traditional minimalist approaches, but also for the crash-proof syntax project as envisaged by Frampton & Gutmann. In order to see this, it must be noted that Dutch does allow NPI's when movement into SpecNegP is blocked for some independent reason, e.g., when the negative clause contains some other negative phrase, as in (12a), or when NegP is situated externally to the clause containing the NPI, as in (12b); see Broekhuis & Klooster (2007) for a more extensive discussion of examples like these.

(12) a Niemand heeft ook maar iets gezien.
Nobody has anything seen
b. Ik denk niet [$_{CP}$ dat Jan ook maar iets gezien heeft]]
I think not that Jan anything seen is

This shows that Frampton & Gutmann cannot appeal to some accidental gap in the Dutch lexicon. Consequently, they can only avoid the use of filters by attributing the contrast between the Dutch and English examples in (6) and (7) to some (so far unknown and unprecedented) language-specific property of the computational system. Again this clashes with the minimalist basic assumption/guiding intuition that the computational system C_{HL} is invariant.

3.3 Other differences

However interesting the minimalist enterprise may be, it is clear that its scope is somewhat limited: it has a lot to say about the restrictions on agreement and movement, but there are many interesting linguistic phenomena that fall outside its scope. A good example of this is the pronunciation pattern of relative clauses in English.

(13) a. *the man [$_{CP}$ who that [I saw t_i yesterday]]
b. the man [$_{CP}$ who ~~that~~ [I saw t_i yesterday]]
c. the man [$_{CP}$ ~~who~~ that [I saw t_i yesterday]]
d. the man [$_{CP}$ ~~who that~~ [I saw t_i yesterday]]

To our knowledge, minimalist research has not paid any attention to this, which need not surprise us given that the computational system C_{HL} does not seem to be equipped to handle data of this sort. Even if one would postulate an operation Delete, which would be able to delete phonological features under certain conditions, it would be very hard to account for the fact that the English pattern in (13) differs from the Dutch pattern in (14) without attributing language-specific properties to the generator. An OT-approach fares much better in this respect as was extensively discussed in Pesetsky (1997/1998), Dekkers & Broekhuis (2000), and Dekkers (1999).

(14) a. *de man [$_{CP}$ die dat [ik gisteren t_i zag]]
b. de man [$_{CP}$ die ~~dat~~ [ik gisteren t_i zag]]
c. *de man [$_{CP}$ ~~die~~ dat [ik gisteren t_i zag]]
d. *de man [$_{CP}$ ~~die dat~~ [ik gisteren t_i zag]]

The fact that the current minimalist approaches, including the one by Frampton & Gutmann, lack the means to account for the patterns like (13) and (14) can, of course, be "solved" by assuming that the phonological component should handle these facts. Taking such a step is, however, quite unsatisfactory given that there is in fact no well-established minimalist theory of the phonological component that could handle such facts; see also footnote 1. Given that data like these (as well as many other phenomena, like verb movement, that have been banished from core syntax over the last fifteen years) can be handled in a perfectly natural way by means of an OT-evaluation, we conclude again that the OT-evaluator is indispensible.

3.4 The universal generator

We want to conclude this section by replying to a reviewer's comment that "[i]t seems more natural to account for [the phenomena discussed above] locally, then to postulate special mechanisms to record that something must happen later on at an interface stage." Unfortunately, we do not really understand how the reviewer has arrived at this conclusion given that we hope to have shown that handling these phenomena locally (= during the derivation) comes at too high a price and that we do not postulate any special mechanism but appeal to a more general theory (the OT-evaluation) about language-specific filters, which, we have argued, are independently needed.

We could leave it to that, but it may nevertheless be useful to say something about the specific example of such a special mechanism mentioned by the reviewer: “elements that must be elided in PF are ‘flagged’ [for deletion; HB&RV] in syntax”. The reviewer seems to prefer such facts to be handled locally by means of a language-specific generator. Actually, we consider the two solutions notational variants and believe both to be wrong for the simple reason that it is very doubtful that ellipsis is part of core syntax as it is currently understood. In order to see this, it must be noted that elision normally depends on the relation between separate sentences (which may or may not be coordinated): in the gapping example in (15a) the finite verb in the second conjunct can be elided under identity with the finite verb in the first conjunct, and in the sluicing example in (15b′), most of the embedded clause can be elided given that its content can be recovered from the preceding question.

(15) a. [[$_{CP}$ Bill kissed Peter] and [$_{CP}$ Mary ~~kissed~~ Jane]] (gapping)
b. Will Bill be around today?
b′. Yes, but I don’t know when exactly ~~Bill will be around today~~. (sluicing)

The phenomena illustrated in (15) are clearly discourse phenomena and therefore resemble the interpretation of the personal pronouns in examples like *John met Bill and he asked him whether...*, about which core syntax hasn’t much to say either: all that the standard theories on binding require is that personal pronouns be free within their binding domain, and the actual interpretation of the pronouns in the second conjunct must be accounted for by appealing to discourse theory. Similarly, we believe it would be unwise to try to account for the facts in (15) by appealing to properties of the generator, irrespective of whether this is done by ‘flagging’ or otherwise. In short, the argument that the reviewer seems to have in mind against our modeling of core grammar is based on the wrong premise that we need to appeal to ‘flagging’.

For completeness’ sake, it should be said that the above conclusion may but need not be incompatible with the claim that elision does take place during the syntactic derivation. But if it does, the *universal* generator would simply randomly produce candidates with and without elision, and it would be the *language-specific* evaluator that selects the optimal candidate. Since at this point we do not have any specific analysis in mind, we will leave the proper treatment of such facts to future research.

3.5 Conclusions

This section has shown that Frampton & Gutmann’s ideal of eliminating filters can only be obtained by attributing language-specific properties to the derivational component, which goes against the minimalist basic assumption/guiding intuition that the computational system C_{HL} is invariant in the sense that it does not exhibit language-specific properties. Furthermore, the elimination of filters requires that C_{HL} be made sensitive to, e.g., pragmatic notions like focus/presupposition, which would mean a radical break with the generative tradition by dropping the postulate of the autonomy of syntax. It may also mean that C_{HL} must be complicated by attributing novel and thus far unprecedented properties to it, which goes against the minimalist goal of reducing the computational system to its absolute minimum. We believe that, from a minimalist point of view, this should be considered too high a price to be paid, and conclude from this that Frampton & Gutmann have set their mark too high by requiring that crash-proof syntax should make the use of filters superfluous.

4 Why developing a crash-proof syntax may be desirable

Saying that Frampton & Gutmann overstated their mark by requiring that crash-proof syntax makes the use of filters superfluous does not automatically mean that their project should be dismissed as meaningless; their more moderate objective of developing a computational system that "generates only objects that are well-formed and satisfy conditions imposed by the interface systems" may be extremely useful for OT-syntacticians that work in the minimalist tradition for reasons that will become clear later in this section.

When one considers the OT-syntax literature so far, it will not go unnoticed that the emphasis is mainly on the OT-evaluator; the analyses provided in this literature virtually never appeal to properties of the generator in order to account for some phenomenon. This is hardly surprising given that the development of OT-syntax can be seen as a reaction to the ever increasing emphasis on the derivational aspects of grammar within the Chomskyan tradition, which is taken to its extreme by the Frampton & Gutmann's crash-proof syntax project. OT-syntax initiated a shift of attention to the representational aspects of grammar (cf. Vogel 2006a), and by doing so provided new means to study empirical domains that were outside the scope of, ignored by, or developed in only an impressionistic manner by the mainstream generative research of the last twenty-five years. We can illustrate this by repeating some of the illustrations given above in somewhat more general terms: Costa (1998), for example, has shown that appealing to OT makes it possible to account for certain correlations between the information structure of the clause and word order, like we find, for instance, in the Icelandic examples in (3) above; Pesetsky (1997/1998) and Dekkers (1999) have shown that some of the filters and principles proposed in the early *principles-and-parameter* period, like the DOUBLY-FILLED-COMP FILTER and the EMPTY CATEGORY PRINCIPLE, can be readily expressed (with more empirical success) by means of a set of more primitive violable constraints; Broekhuis (2008) has argued that the OT-evaluator can be used to provide a general format for expressing the minimalist interface conditions, which were so far often formulated in an impressionistic and random way.

Given this shift in attention and its initial success, it need not surprise us that people soon started to investigate whether certain properties that were previously attributed to the generator could also be handled by the evaluator: cf. the discussion in Section 2. It is our believe that this is not something that the minimalist audience should look upon with suspicion, given that this might make it possible to pursue the minimalist goals with even more success: see the discussion in Section 3.1, which has shown that adopting the Effect-on-Output Condition from Chomsky (2001) or its OT-counterpart developed in Broekhuis (2008) enables us to eliminate the epp-features, which are actually quite outlandish from a minimalist point of view. This elimination of the epp-feature should also be applauded given that it enables us to develop a version of the computational system that is truly invariant for all natural languages. Of course, this line of research may eventually lead to an even further reduction of C_{HL}, that is, lead to a more representational system like the one hinted at in Vogel (2006a), which, in effect, would lead linguistic theory into the opposite direction of Frampton & Gutmann's crash-proof syntax project.

The previous paragraph suggests that, at this point, current linguistic theories can be placed on a continuum ranging from purely derivational to purely representational. For example, the crash-proof syntax project aims at developing a theory that is placed at the left-hand border of the continuum in Figure 2, the simple-generator approach of Vogel (2006a) is placed at the right-hand border, and the derivation-and-evaluation approaches developed in Dekkers (1999) and Broekhuis (2008) occupy some intermediate position.

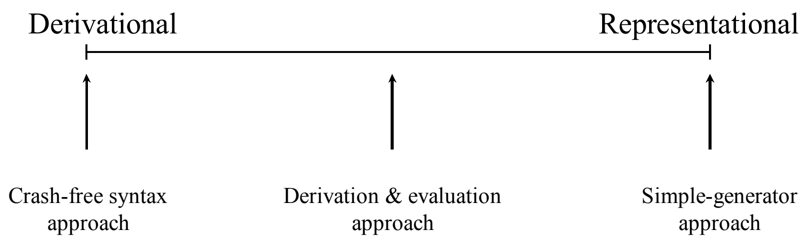

Figure 2: continuum of syntactic approaches

All approaches on this continuum (whether they be based on mainstream generative syntax or not) may pursue the same minimal goals of reducing grammar to its absolute minimum. The crash-proof syntax approach aims at reducing the grammar to the derivational component by eliminating all conditions or filters on the output representations. The simple-generator approach, on the other hand, aims at reducing the role of the generator and places the explanatory burden on the OT-evaluator by focusing on input-output correspondences (akin to what we find in phonology). The derivation-and-evaluation approach explicitly takes an intermediate position and claims that the explanatory burden must be placed both on the generator and the evaluator by reducing both components to their minimum: components that are alien to the generator, like the epp-features, will be eliminated when it is possible to account for them by postulating some violable constraint of sufficient generality; at the same time, the postulated constraints will be kept to a minimum by attributing to the generator what comes to it naturally. The more limited goal of Frampton & Gutmann's crash-proof syntax project fits in quite well with the latter approach: when the generator only produces converging representations, we can avoid the introduction of constraints that are only needed to eliminate the crashing derivations. At this point, it seems impossible to predict whether the fully derivational, the fully representational, or some mixed approach will lead to the best results, but this paper has contributed to the discussion of what type of approach is superior by pointing out some, in our view, inherent problems for the fully derivational approach.

5 Conclusion

This article has argued that it is very hard, if not impossible, to pass judgment on Frampton & Gutmann's crash-proof syntax project from a more general optimality-theoretic perspective. This is due to the fact that in some OT-approaches to syntax, the notion of (convergent) derivation does not play a role. The only thing we could do, therefore, was to discuss Frampton & Gutmann's claim that the notion of filter should be eliminated from the theory, as this goes against the guiding intuition of OT that the language system contains a filtering device that determines which candidates from the Candidate Sets produced by the generator are optimal in a given language. We believe to have shown that the elimination of filters is not possible within the Agree-based version of the minimalist program adopted by Frampton & Gutmann without running foul with the minimalist basic assumption/guiding intuition that the computational system C_{HL} is invariant. We have further argued that the more modest goal of the crash-proof syntax project of developing a system that "generates only objects that are well-formed and satisfy conditions imposed by the interface systems" may be welcome in OT-approaches that adopt some version of C_{HL} as the generator.

References

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

Broekhuis, Hans. 2006. Derivations (MP) and evaluations (OT). In *Optimality Theory and Minimalism: A Possible Convergence? Linguistics in Potsdam 25*, eds. Hans Broekhuis and Ralf Vogel, 137–193. University of Postdam: http://www.ling.uni-potsdam.de/lip.

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

Broekhuis, Hans, and Joost Dekkers. 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.

Broekhuis, Hans, and Wim Klooster. 2007. Merge and Move as costly operations. In *Groninger Arbeiten zur germanistischen Linguistik*, ed. Jan-Wouter Zwart, 17-37: http://gagl.eldoc.ub.rug.nl/root/Volume45/broekhuisklooster/.

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.

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

Costa, João. 1998. Word order variation. A constraint-based approach, University of Leiden/LOT: Ph.D. dissertation.

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

Frampton, John, and Sam Gutmann. 2002. Crash-proof syntax. In *Derivations and explanations in the minimalist program*, eds. Samuel David Epstein and T. Daniel Seely, 90-105. Malden (UsA).Oxford (UK), Blackwell Publishing.

Frampton, John, and Sam Gutmann. 2006. How sentences grow in the mind. In *Agreement systems*, ed. Cedric Boeckx, 121-157. Amsterdam, John Benjamins.

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.

Haegeman, Liliane. 1995. *The syntax of negation*. Cambridge: Cambridge University Press.

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.

Holmberg, Anders. 1986. Word order and syntactic features in the Scandinavian languages and English, University of Stockholm, Department of General Linguistics: Ph.D. dissertation.

Holmberg, Anders. 1999. Remarks on Holmberg's generalization. *Studia Linguistica* 53:1-39.

Moro, Andrea. 1997. *The raising of predicates. Predicative noun phrases and the theory of clause structure*. Cambridge: Cambridge University Press.

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.

Neeleman, Ad. 1994. Complex predicates, University of Utrecht: Ph.D. dissertation.

Pesetsky, David. 1997. Optimality theory and syntax: movement and pronunciation. In *Optimality 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.

Vikner, Sten. 2001. The Interpretation of Object Shift and Optimality Theory. In *Competition in Syntax*, eds. Gereon Müller and Wolfgang Sternefeld, 401-442. Berlin, Mouton de Gruyter.

Vogel, Ralf. 2004. Correspondence in OT Syntax and Minimal Link Effects. In Minimality Effects in Syntax. In *Minimality Effects in Syntax*, eds. Gisbert Fanselow, Arthur Stepanov and Ralf Vogel, 401-442. Berlin, Mouton de Gruyter.

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.

Vogel, Ralf. 2006b. Weak Function Word Shift. *Linguistics* 44:1059-1093.

Woolford, Ellen. 2007. Case Locality: Pure Domains and Object Shift. *Lingua* 117:1591-1616.