Erschienen in: Measuring Grammatical Complexity / Frederick J. Newmeyer ... (Hrsg.). - Oxford : Oxford University Press, 2014. - S. 128-147. - ISBN 978-0-19-968530-1

The complexity of narrow syntax: Minimalism, representational economy, and simplest Merge

ANDREAS TROTZKE AND JAN-WOUTER ZWART

7.1 Introduction

The issue of linguistic complexity has recently received much attention by linguists working within typological-functional frameworks (e.g. Miestamo, Sinnemäki, and Karlsson 2008; Sampson, Gil, and Trudgill 2009). In formal linguistics, the most prominent measure of linguistic complexity is the Chomsky hierarchy of formal languages (Chomsky 1956), including the distinction between a finite-state grammar (FSG) and more complicated types of phrase-structure grammar (PSG). This distinction has played a crucial role in the recent biolinguistic literature on recursive complexity (Sauerland and Trotzke 2011). In this chapter, we consider the question of formal complexity measurement within linguistic minimalism (cf. also Biberauer *et al.*, this volume, chapter 6; Progovac, this volume, chapter 5) and argue that our minimalist approach to complexity of derivations and representations shows similarities with that of alternative theoretical perspectives represented in this volume (Culicover, this volume, chapter 8; Jackendoff and Wittenberg, this volume, chapter 4). In particular, we agree that information structure properties should not be encoded in narrow syntax as features triggering movement, suggesting that the relevant information is established at the interfaces. Also, we argue for a minimalist model of grammar in which complexity arises out of the cyclic interaction of subderivations, a model we take to be compatible with Construction Grammar approaches. We claim that this model allows one to revisit the question of the formal complexity of a generative grammar, rephrasing it such that a different answer to the question of formal complexity is forthcoming depending on whether we consider the grammar as a whole, or just narrow syntax. The grammar as a whole, including interface components in addition to narrow syntax, as well as recursive interaction among subderivations, is vastly more complicated than a finite-state grammar, but there is no reason for concluding that narrow syntax is not simply finite-state.

The chapter is structured as follows. In section 7.2, we clarify the generative perspective on measuring linguistic complexity and distinguish this approach from performance-oriented notions that are traditionally confused with the generative account. Section 7.3 shows how complexity reduction in the domain of syntactic representations results in points of convergence between minimalism and other perspectives that assume that linguistic complexity does not arise from syntax alone. In section 7.4, we turn to the layered-derivation perspective on linguistic complexity and argue that narrow syntax can be captured by a finite-state device and, therefore, falls low on the Chomsky hierarchy. Section 7.5 summarizes the main results of the chapter, followed by a short conclusion.*

7.2 The generative perspective on measuring linguistic complexity

Within the generative paradigm, the question of the comparative complexity of languages, as discussed in recent typological-functional literature, does not arise (cf. also Sauerland 2014). It has been a core assumption of the generative research program since its beginnings that the complexity of individual languages is determined by the invariant biological mechanisms underlying human language in general. Of course, a more differentiated picture might emerge when we look at the interconnection of specific parameters that are set in different languages and assume that there is some complexity metric that classifies parametric 'routes' to certain grammars as simpler than routes to certain other grammars (Biberauer *et al.* this volume, chapter 6). However, within generative linguistics, such an approach is not uncontroversial, since the theory of grammar, as currently understood, has no room for connecting statistical generalizations to properties of the faculty of language in the narrow sense ('I-language') or to any principles and parameters associated with I-language.[1] As Newmeyer (2007: 240) points out, "correlations between complexity and rarity are not [...] to be expected, since implicational and frequency-based generalizations do not belong to the realm of I-language." In this chapter, we do not

* We thank the audiences at the workshop on *Formal Linguistics and the Measurement of Grammatical Complexity* (23–24 March 2012, University of Washington, Seattle) and at the workshop on *Complex Sentences, Types of Embedding, and Recursivity* (5–6 March 2012, University of Konstanz). We are especially grateful to the editors of this volume, Fritz Newmeyer and Laurel Preston, for the careful discussion of every aspect of this chapter. All remaining errors and shortcomings are our own. Andreas Trotzke gratefully acknowledges financial support from the DFG Excellence Initiative (University of Konstanz, project no. 65411).

[1] In this chapter, we equate I-language with the faculty of language in the narrow sense as defined in Hauser, Chomsky, and Fitch (2002), i.e. the component of the grammar that applies simple rules merging elements, also referred to as 'narrow syntax' below.

delve into these issues. Instead, we focus on basic derivational and representational aspects of linguistic complexity that are not subject to variation.[2]

If we concentrate on I-language, the issue of measuring complexity among natural languages disappears. What does not disappear, however, is the question where the grammar of natural language falls on the complexity hierarchy of *formal* languages. In the 1950s, Chomsky showed that a particular type of recursion is essential to drawing the line between the phrase structure models of language that he proposed and models of language prevalent in contemporary structuralist thinking. In particular, Chomsky (1956, 1959) showed that self-embedding involves the kind of complexity that requires (at least) context-free grammars, rather than less complex types of grammar (specifically, finite-state devices). Chomsky (1959: 148) defined this notion of self-embedding as follows (*I* is the identity element, i.e. zero, and $\Rightarrow$ indicates a derivation involving rewrite operations):

(1) A language L is *self-embedding* if it contains an A such that for some φ, ψ ($\varphi \neq I \neq \psi$), $A \Rightarrow \varphi A \psi$.

The definition characterizes as self-embedding any language that contains a string A and allows the derivation from A of a string that properly contains A, that is, A is preceded and followed by two non-trivial strings. Chomsky (1957) went on to show that patterns such as (2) exist in English (slightly modified from Chomsky 1957: 22, with *It's raining* as the declarative sentence S). These patterns clearly satisfy the definition of self-embedding in (1):

(2) a. S $\Rightarrow$ If S, then it's true.
b. S $\Rightarrow$ If it's raining, then it's true.
c. S $\Rightarrow$ If if it's raining, then it's true, then it's true.
d. (...)

As Chomsky notes, S in (2a) can in fact have the same structure as the sentence to the right of the arrow in (2a). As a result, the end product of the derivation may be a string with a mirror image pattern ($if_1 \ldots if_2 \ldots if_n \ldots then_n \ldots then_2 \ldots then_1$). This mirror image pattern cannot be generated by a finite-state grammar, since this device

[2] A point of clarification is in order at this point. Our discussion in this chapter focuses on narrow syntax, a core component of the model of grammar, but not the only component. In particular, the model includes interface components dealing with sound and meaning. For a fuller treatment, the question of the complexity of the grammar has to be answered separately for the grammar as a whole and for the individual components (including narrow syntax), with different answers forthcoming in each case. It is also an open question which phenomena are to be associated with which component of the grammar, with current proposals relocating seemingly narrow syntactic phenomena such as head movement and inflectional morphology to the interface component dealing with sound (e.g. Chomsky 2001). We abstract away from these questions and proceed on the understanding that narrow syntax involves nothing more than a sequence of operations Merge, joining elements from a predetermined set (the Numeration) into a hierarchical phrase structure.

computes a string strictly locally and thus cannot ensure an equal number of *if*s and *then*s (see Chomsky 1956 for a more rigorous presentation of the argument).

The relation between formal grammar and processing complexity had been addressed extensively by Chomsky and Miller's (1963) seminal work. By referring to the property of recursive self-embedding, they argued in favor of drawing a sharp distinction between processes at the level of performance and mechanisms at the level of formal grammar. As is well known, their observation was that multiple center-embedding leads to structures that cannot be produced or comprehended under normal on-line conditions, as illustrated by (3):

(3) The rat the cat the dog chased killed ate the malt.
(Chomsky and Miller 1963: 286)

Chomsky and Miller argued that the fact that such sentences are quite incomprehensible has no bearing on the desirability of generating them at the level of formal grammar, because, as Chomsky (1963: 327) pointed out by means of an analogy, "the inability of a person to multiply 18,674 times 26,521 in his head is no indication that he has failed to grasp the rules of multiplication." In other words, such structures are more complex than others due to performance factors that limit the realization of our grammatical competence. In response, however, Reich (1969) was the first to propose an FSG capable of generating sentences with degree-1 center-embedding but not center-embeddings of degree 2 or higher. Recently, approaches similar to those of Reich have been pursued in a connectionist setting by Christiansen and Chater (1999) and Christiansen and MacDonald (2009).[3] These accounts not only argue that natural languages are not of a PSG-type; they also claim that complexity measurement according to the Chomsky hierarchy in general is not motivated. They observe that self-embedding structures of a certain degree are not attested in linguistic performance and therefore argue that they should not be generable by the grammar.

The crucial distinction between these 'performance-oriented accounts' and the generative approach to complexity measurement is very clear. According to the generative perspective, the performance limitations on recursive self-embedding are captured by factors extrinsic to the competence grammar (such as memory overload induced by distance, cf. Gibson and Thomas 1999; Gibson 2000). In contrast, performance-oriented accounts such as usage-based approaches claim that "constraints on recursive regularities do not follow from extrinsic limitations on memory or processing; rather they arise from interactions between linguistic experience and architectural constraints on learning and processing [. . .] intrinsic to

[3] In section 7.4, we argue that center-embeddings can also be generated by a finite-state device. However, while connectionist approaches and their precursors do not assume a mentally represented grammar that allows unbounded recursion, we propose a competence grammar that allows generating infinite center-embeddings.

the system in which the knowledge of grammatical regularities is embedded" (Christiansen and MacDonald 2009: 127). In other words, while the generative approach postulates a competence grammar allowing unbounded recursion, the performance-oriented accounts deny the mental representation of infinite recursive structure and, thereby, try to nullify one of the axioms of modern linguistic theory: the grammar–performance distinction. A detailed discussion of these two positions would take us too far afield here (for a more general discussion, see Newmeyer 2003).

In this chapter, we follow recent work by Trotzke, Bader, and Frazier (2013) and Trotzke and Bader (2013), who present empirical evidence in favor of the grammar–performance distinction in the context of recursive self-embedding. Accordingly, in our view, the measurement of computational complexity, as represented by the Chomsky hierarchy, cannot be fruitfully connected to performance complexity, in keeping with the arguments of Chomsky and Miller (1963). Instead, the generative perspective on measuring linguistic complexity both abstracts away from linguistic variation and from processing complexity and focuses on basic formal notions of computational complexity. This is most clearly evidenced in the most recent version of generative grammar, namely the Minimalist Program (MP).

According to Chomsky (1995: 221), the MP is "a research program concerned with [...] determining the answers to [...] the question: 'How 'perfect' is language?'" In other words, the MP explores the hypothesis that language is a system that meets external constraints imposed by other cognitive components in the most 'elegant' (read: economical) way. Accordingly, as pointed out by Wilder and Gärtner (1997), within the MP, 'economy' is not only understood as a methodological postulate concerned with providing the 'simplest' description of a linguistic phenomenon. Rather, economy is also understood as referring to a property of language itself. Given this notion of the human language faculty, computational complexity arguments in terms of 'least effort' metrics play an important role in linguistic minimalism (e.g. Chomsky 1991; Collins 1997).

Let us briefly turn to basic computational aspects of minimalism in order to demonstrate in what sense they can be regarded as computationally 'optimal.' When we turn to computational science, two basic components of an algorithmic procedure must be distinguished: its time complexity and its space complexity (for a more extensive discussion of what follows, cf. Manber 1989; Mobbs 2008). While the number of operations required to perform a specific task constitutes the time complexity of an algorithm, the amount of working memory required for the performance of a task represents its space complexity. Accordingly, reducing both the time and the space complexity of running an algorithm results in more computational optimality. Now, how do minimalist conditions on derivations and representations correspond to these fundamental concepts of computational complexity theory?

To begin an answer to this question, let us look at the Extension Condition, proposed by Chomsky (1995). This condition on syntactic derivation holds that

"Merge always applies in the simplest possible form: at the root" (Chomsky 1995: 248), that is, there is only one possible site at which to extend a phrase marker. This condition thus minimizes complexity in accordance with fundamental complexity metrics of computational science. As Mobbs (2008: 29) points out, postulating "more than one possible site at which to Merge, it would be necessary to search for the appropriate site in each case, increasing the operational load [= the time complexity, AT/JWZ] on computation." The general constraint on syntactic derivations that ensures that the system meets such abstract complexity measures is the assumption that natural language syntax, as understood in the minimalist sense sketched above, should be operating in a maximally economical way.

Non-minimalist frameworks such as Simpler Syntax (Culicover and Jackendoff 2005, 2006) have also addressed the problem of the measurement of grammatical complexity. For example, this theory assumes that "the complexity of syntactic structure involves the extent to which constituents contain subconstituents, and the extent to which there is invisible structure" (Culicover and Jackendoff 2006: 414). In particular, this account attributes a higher syntactic complexity to 'mainstream generative grammar' (Culicover and Jackendoff's term), since mainstream approaches, as understood in the Simpler Syntax framework, operate with covert levels of representation like 'Deep Structure' and 'Logical Form' and with 'invisible' elements in the syntactic tree. These 'extra' levels and elements increase the representational complexity of syntactic structures, as Culicover and Jackendoff (2006: 413) briefly point out in the context of control constructions. According to their view, the situation that *drink* in *Ozzie tried not to drink* does have a subject does not necessitate an account that postulates 'hidden' syntactic representations like 'PRO.' In contrast, the interpretation of *Ozzie* as the 'drinker' can be formulated as a principle of semantic interpretation, external to the syntactic component.[4] So, according to Jackendoff (2008: 197), the main critique of minimalism refers to its complex representational format, since minimalism "requires null elements, a covert level of syntax, and particular hitches in the syntax that correlate in theoretically dubious fashion with the semantic peculiarities of the constructions in question."

In section 7.3, we show how the minimalist approach to these aspects of the representational format partly converges with the Simpler Syntax model. Since information structural properties like focus and topic are often accounted for in terms of an enriched syntactic representation and covert syntactic operations like LF movement (cf. Chomsky 1976 and later work), we will turn to this issue. In section 7.4, we discuss derivational complexity, i.e. the complexity of the structure building

[4] Interestingly, with regard to the very same phenomenon, recent minimalist literature argues in a similar vein. For example, Hornstein, Nunes, and Grohmann (2005: 54) replace the Theta-Criterion, which operates at Deep Structure, with a 'Theta-Role Assignment Principle' that applies at the semantic interface (LF).

process that generates syntactic representations, and argue for points of convergence with Construction Grammar approaches.

7.3 Representational complexity, cyclicity, and Simpler Syntax

In this section, we focus on representational complexity and its measurement in linguistic minimalism and in the Simpler Syntax framework. Advocates of the latter have claimed that "[t]he Minimalist Program [. . .] assumes that the structures and derivations of Principles and Parameters Theory are essentially correct" (Culicover and Jackendoff 2005: 88). They go on to argue that the elaborate and abstract structures of Principles and Parameters Theory are to be discarded in favor of considerable reduction of the representational format. In Simpler Syntax, this is achieved by relegating a substantial body of phenomena to 'interface rules.' In what follows, we demonstrate that proponents of a "constraint- and construction-based minimalism" (Jackendoff 2008: 222) do not fully acknowledge the recent shift from a representational to a strong derivational theory of linguistic structure within the MP. We argue that, given this shift, which involves minimizing the representational format of the computational system, the recent 'dynamic' approaches within the MP share basic assumptions with the perspective advocated under the Simpler Syntax hypothesis. To illustrate, we focus on an issue that has frequently been discussed with respect to representational complexity: the representation of information structural notions in syntactic structure. But before turning to this particular issue, we first point out the basic characteristics of the dynamic derivational model assumed in the MP, where cyclic (phase-based, layered) computations play a central role.

7.3.1 *Redefining derivations vs. representations*

The idea that the computation of a syntactic structure proceeds phase by phase (Chomsky 2000 and later work) has important consequences for the representational configuration, since the minimalist model of the grammar no longer defines a single point of 'Spell-Out' handing an entire syntactic structure to the interface components LF (the semantic component) and PF (the articulatory component). Instead, each derivation contains multiple points of interaction between the syntactic component (narrow syntax) and the interface components (LF and PF), the actual number depending on the number of phases. This dynamic interaction with the interfaces is also true of the model proposed in Zwart (2009a, section 7.4), where phases are redefined as derivation layers, that is, finite sequences of Merge yielding a substructure to be processed by the interface components and to be used in a further (sub) derivation. In a model with phases or derivation layers, the derivation of a sentence is a *system of derivations*, with multiple subderivations each feeding the interface components separately. So, while syntactic structures were hitherto considered to represent the whole sentence at some particular level of representation, it is now argued that derivations consist of stages in which only parts of these structures are represented at the interfaces. Put differently, "while there are still what might be called PF and LF components, there are no *levels* of PF and LF" (Lasnik 2005: 82, emphasis in the original). Consequently, the model of grammar resulting from these considerations can in effect be regarded as 'level-free,' as also pointed out by Boeckx (2006: 77).

The basic intuition behind models with multiple Spell-Out is that 'chunking' the derivation in subderivations leads to a reduction of computational complexity. As Miller (1956) had shown in the context of limitations on the amount of pieces of structure needed to be held in memory, "[b]y organizing the stimulus input simultaneously [. . .] into a sequence of chunks, we manage to break [. . .] this informational bottleneck" (Miller 1956: 95). Using phases or derivation layers in linguistic computation builds on this result. Otherwise put, and referring back to basic computational aspects sketched in section 7.2, chunking the derivation in phases or layers reduces the space complexity by reducing the amount of working memory required for running the algorithm (cf. Chesi 2007 and his specification of complexity reduction in phase theory).

Moreover, phase-based or layered derivations lead to a less redundant system by reducing the number of independent cycles in the computation. More concretely, in Principles and Parameters theory (Chomsky 1981), there were at least three relatively independent generative systems, all operating on the same domain, but separately. In Chomsky's (2004a: 151) words, "[t]here was one that formed d-structure by X-bar Theory, which is basically cyclic. There's the transformational cycle, which is mapping d-structure to s-structure. There's a covert transformational cycle, which is mapping s-structure to LF, with the same kinds of transformations and also cyclic." Chomsky claims that a phase-based derivation, with its cyclic transfer property, comes closer to the ideal of a single-cycle architecture.

One consequence of reducing the complexity of the model of the grammar in the context of levels of representations and covert syntactic operations remains to be explored. Given the dynamic interaction with the interfaces postulated in this 'strong' (i.e., in effect, 'level-free') derivational approach (cf. Brody 2002 for different incarnations and 'strengths' of derivational theories), it has been argued that this model allows for a direct interaction between PF and LF, that is, "PF has access to both, the syntactic derivation of the phase [. . .] and the semantic interpretation" (Winkler 2005: 24). As is well known, this direct interaction between the phonological and the semantic component is a crucial feature of Jackendoff's (1997 *et seq.*) Parallel Architecture, which is the model of grammar assumed in Simpler Syntax. This point of convergence between minimalism and Simpler Syntax has, to our knowledge, so far been mentioned only by Winkler (2005), who notes that the strong derivational model within minimalism "turns out to be conceptually closer to Jackendoff's [. . .] tripartite parallel model of grammar than might be recognized at first sight" (Winkler 2005: 231, n. 8). However, she does not elaborate on this point and leaves it to a short comment in a footnote.

7.3.2 *Reducing representational complexity: a minimalist perspective on the syntax-pragmatics interface*

Let us bring down the comparison of recent minimalism and the Parallel Architecture model to tractable size by focusing on the analysis of one specific phenomenon, namely the pragmatics of left-periphery-movement (LP-movement) in German, a topic that is described in information structural terms in the literature. Ray Jackendoff has repeatedly argued that, especially in the context of information structure, "there are aspects of semantics that have no impact on syntax but *do* have an effect on phonology" (Jackendoff 2003: 658, emphasis in the original). Accordingly, following Jackendoff's argument, cases like focus expressed through prosody seem to require a direct phonology–semantics interface, where "interface principles [. . .] map directly between a string of phonological words and a meaning" (Jackendoff and Wittenberg, this volume, chapter 4).

In an influential version of the standard generative framework, the 'cartographic program,' information structural properties of sentences are accounted for by encoding properties of information structure in the syntactic representation (e.g. Rizzi 1997). Thus, these accounts consider information structural properties to be represented in the syntax as formal categories actively determining the syntactic derivation. However, as cases like phonologically expressed focus demonstrate best, such a syntactic feature seems to provide no more than a device to pass information from semantics to phonology. Accordingly, Jackendoff (2002: 409) concludes that such a syntactic feature "is simply an artifact of syntactocentrism, the assumption that everything in meaning has to be derived from something generated in syntax." We now argue that the goal of reducing representational complexity in minimalism is consistent with Jackendoff's view. Before going into detail here, let us point out that we do not reject the *descriptive* advantages of the cartographic framework. Approaching syntactic structures (and especially the clausal left periphery) from a cartographic perspective has proven to be incredibly fruitful. Since proponents of this approach are committed, by and large, to a rigorous methodology of description, they can rely on a large amount of previous work and thereby also refine our picture of the overall syntactic structure of heretofore under-researched languages (for this point, cf. Trotzke 2012). However, in this chapter, we are concerned with the complexity of *narrow* syntax—a domain that refers to universal computational properties of natural languages and thereby contributes to the achievement of *explanatory* adequacy. These two goals, of course, complement each other; as Rizzi (2013b: 213) points out, "an accurate map of the sequence [of functional projections] is the essential point of departure for further study, including the search for further explanation" (cf. also Ramchand and Svenonius 2013 in this regard). However, in the domain of explanatory adequacy, it can be argued that even 'weak versions' of the cartographic account are incompatible with the dynamic approach to syntactic structure assumed in some implementations of the MP (cf. Zwart 2009b). In accordance with Newmeyer (2009: 131, emphasis in the original), we would therefore like to argue "that oceans of functional projections on the left periphery represent a singularly *unminimalist* framework for capturing [...] variation. Given that few languages manifest the proposed cartography in its fullness, there is no benefit to proposing that UG provides the set of projections and their ordering."

However, according to the proponents of the cartographic approach, and in contrast to Simpler Syntax and also to our view, there is no tension between enriched representations as proposed in cartography and minimalist ideas of reducing the syntactic computation to a minimum. In particular, as Rizzi (2004) argues, a crucial point of connection between the cartographic approach and the MP is the core idea of computational simplicity. The cartographic approach, according to Rizzi, contributes to this notion by decomposing functional projections into simple structural units. Thus, regarding complexity measurement, "[l]ocal simplicity is preserved by natural languages at the price of accepting a higher global complexity, through the proliferation of structural units" (Rizzi 2004: 8). However, cartographic approaches, with their enriched syntactic representations, are at root incompatible with the strongly derivational assumptions of standard minimalism. To be specific, proponents of the cartographic approach postulate functional heads in the left periphery possessing designated features for focus, topic, and other information structural constructs. According to Chomsky's (1995: 228) economy conditions, however, "any structure formed by the computation [...] is constituted of elements already present in the lexical items selected [...]; no new objects are added in the course of computation apart from rearrangements of lexical properties." In other words, this 'Inclusiveness Condition' implies that syntactic operations can refer only to lexical features. But of course, lexical items cannot be viewed as *inherently* possessing information structure properties. Consequently, such properties, as Neeleman and Szendrői (2004: 155) note, "must be inserted after an element has been taken from the lexicon," and thus the postulation of discourse-oriented features and the functional heads hosting them violates the Inclusiveness Condition.

Let us now turn to an empirical phenomenon in German that challenges cartographic accounts and their consequent higher global complexity. Recall that, according to Culicover and Jackendoff's (2006: 414) complexity measure, "the complexity of syntactic structure involves the extent to which constituents contain subconstituents, and the extent to which there is invisible structure." In what follows, we show how the German data can be accounted for from a minimalist perspective in a way that both reduces global complexity and is quite compatible with the Simpler Syntax approach.

In German, given certain pragmatic conditions, parts of idiomatic verb phrases can show up in the left periphery (for discussion of more phenomena that involve similar issues as discussed in this section, cf. Trotzke 2010). Consider the following example (for similar cases, cf. Fanselow 2004: 20):[5]

(4) den Löffel abgeben ('to die,' lit. 'the spoon pass')
[Den LÖffel]$_i$ hat er t$_i$ abgegeben.
the spoon has he passed
'He died.'

In (4), *den Löffel* receives pitch accent, indicating a contrastive interpretation. However, the preposed part *den Löffel* of the idiom *den Löffel abgeben* is meaningless in isolation (i.e. there is no set of alternatives to *den Löffel* in this particular context). Accordingly, fronting this element challenges the assumption of a dedicated syntactic position in the left periphery associated with focal or contrastive interpretation of the element that occupies this position, since, as Fanselow (2003: 33) puts it, "[i]t is difficult to accept the idea that a meaningless element can be interpreted as a focus or a topic [...] phrase." Thus, according to Fanselow's (2003) terminology, *den Löffel* is moved to the left periphery as a 'pars-pro-toto,' since the pragmatic interpretation involved is equivalent to preposing the whole idiomatic constituent, as shown in (5), where the whole predicate is fronted and interpreted contrastively.

(5) [Den LÖffel abgegeben]$_i$ hat er t$_i$.
the spoon passed has he
'He died (and did not survive).'

The fronting in (4), involving subextraction out of an idiomatic string, is problematic for any restrictive account of syntactic displacement. Even so, it appears to us that a cartographic approach, involving focus heads triggering LP-movement, does not provide a suitable model for the analysis of this phenomenon, since moving only a part of the focus hardly suffices to check a corresponding focus feature. In other words, we agree with Jackendoff's objection to enriching representational complexity by encoding information structure concepts in the narrow syntax.

In what follows, we propose a strongly-derivational minimalist approach that abstracts away from language-specific representations and also implements the idea that "a direct phonology-semantics interface [...] is attractive for the correlation between prosody and information structure" (Jackendoff 2003: 658). Thereby,

[5] In this and the following examples, capitals indicate relatively high pitch.

in accordance with Simpler Syntax, we aim at a considerable reduction of the syntactic representational format.

7.3.3 *A minimalist account of topic/focus interpretation*

We have seen that LP-movement to a topic or focus projection is not triggered by a topic or focus feature in a functional head in the left periphery, and that the topic or focus reading cannot be associated with elements at the point in the derivation where they are merged. Let us consider, then, an alternative approach in which topic or focus interpretation is an emerging feature, arising in the course of a derivation. In such a view, narrow syntax is oblivious to information structure; topic or focus readings are established only at the interfaces.

One way to approach the derivation of information structural interpretations is to say that Merge creates a structural dependency (or a set of such dependencies) that can be interpreted at the interface components in various ways (e.g. in terms of subject-predicate, topic-comment, or focus-ground oppositions, cf. Zwart 2009a: 170). From this perspective, the question of whether fronted elements appear in the left periphery via internal Merge (movement) or external Merge ('base-generation') is moot. What matters is that merging a left-peripheral element sets up a particular dependency allowing for a limited range of interpretations.[6] And just as Merge is not concerned with the notions of topic and focus, it is arguably not concerned with the corresponding prosodic effects. The fact that these effects nevertheless occur suggests that the interpretation established at LF (the semantic component) informs the relevant components of PF (the articulatory component) responsible for clausal prosody. This direct LF–PF interaction, however, does not in our opinion call for a radical revision of the model of grammar, as advocated by Jackendoff. Rather, the direct sound-meaning pairing results from a restrictive interpretation of the role of narrow syntax within the minimalist architecture.

To see that this interpretative approach to topic and focus interpretation is not unconstrained, consider once again the examples of LP-movement in German featuring the idiom *den Löffel abgeben* in (4) and (5). As we noted, LP-movement can front *den Löffel* alone, as in (4), or it can front the entire idiomatic expression *den Löffel abgegeben*, as in (5). The fact that *den Löffel abgeben* is a (noncompositional) idiom requires that the semantic component assembles the split parts of the idiom in (4), creating the idiomatic reading on the fly, as it were. We assume that this interpretive operation is a function of the composition of the two categories *den Löffel* and *hat er abgegeben* (cf. (4)), merged together at the final step of the narrow syntactic derivation. That is, for the correct interpretation of *abgegeben* in (4), the

[6] In what follows, we use the term 'LP-movement' to refer to a particular syntactic construction rather than to a syntactic operation.

interpretation needs to be supplied with *den Löffel*, which can be accomplished in the semantic component, once *den Löffel* has been merged to *hat er abgegeben* (again abstracting away from the question of whether *den Löffel* was moved or base-generated in left-peripheral position).

Let us make our analysis more concrete and state that the object in (6a), which is a constituent structure representation of (4) in the semantic component, has one realization in which the nonidiomatic material is ignored, as in (6b), so that just the idiomatic elements remain (cf. (6c)):

(6) a. (den Löffel) (hat er abgegeben)
b. (den Löffel) (~~hat er~~ abgegeben)
c. den Löffel abgegeben

Informally, we can 'recognize' *den Löffel abgegeben* (6c) in *den Löffel hat er abgegeben* (6a), that is, we may take *den Löffel (...) abgegeben* (6a/c) as a token of the type *den Löffel abgeben* (cf. (4)). The operation in (6b) is allowed, we suggest, since the structural dependency between *den Löffel* and *abgegeben* is the same in the split and the unsplit idiom, with *den Löffel* merged with either *abgegeben* or a constituent containing *abgegeben*.

Observe now that splitting the idiom by fronting *abgegeben* is not possible:

(7) # [Abgegeben]$_i$ hat er den LÖffel t_i
passed has he the spoon
(intended: same as (4–5))

From our perspective, this impossibility follows from the sensitivity of the semantic component to the reversed order of the idiom parts. To establish the idiom on the fly, *abgegeben* would need to be supplemented by *den Löffel* just as in (4), but the required structural dependency is not there, as *den Löffel* is not merged with a constituent containing *abgegeben*. In other words, the stripping procedure that worked well in (6), allowing the interpretation of (6b) as a token of the type (6c), now yields no result, as (8b) differs too much from (8c) for (8b) to be interpreted as a token of the type (8c):

(8) a. (abgegeben) (hat er den Löffel)
b. (abgegeben) (~~hat er~~ den Löffel)
c. den Löffel abgegeben

On this analysis, the ungrammaticality of (7), where *abgegeben* is fronted alone, suggests a constraint not on LP-placement, but on idiom interpretation. Indeed, when idiomaticity is not a factor, constructions like (7) are possible, with the relevant context provided:

(9) [Abgewaschen]$_i$ hat er den LÖffel t_i (und abgetrocknet das Messer)
washed has he the spoon (and dried the knife)
'He washed the spoon.'

Returning to the main theme, we pointed out conceptual points of convergence between recent minimalism and the Simpler Syntax model in light of a derivational analysis of LP-movement in German. In particular, in contrast to approaches that are "forced to generate [...] sentences with a dummy syntactic element such as [+Focus], which serves only to correlate phonology and meaning" (Jackendoff 2003: 658), we demonstrated that information structural interpretation, from a minimalist point of view, emerges in the course of a derivation and thus is established only at the interfaces. In particular, the contrastive interpretation of the whole predicate *den Löffel abgeben* in both (4) and (5) is established at LF, which, in our model with direct LF–PF interaction, informs the relevant components of PF responsible for clausal prosody. This result is in line with the direct sound-meaning interaction postulated in Simpler Syntax, but it is also in accordance with other recent minimalist accounts that claim that "notions of information structure play no role in the functioning of syntax [...]. There is no reason left for coding information structure in the syntactic representation" (Fanselow and Lenertová 2011: 205). Our analysis, with its reduction of representational complexity, is thus preferable in the light of the representational complexity measure sketched by Culicover and Jackendoff (2005, 2006) that attributes a higher complexity to models operating with covert levels of representation and enriched syntactic trees. Note that excluding notions like focus or topic from the syntactic representation also avoids, to our mind, unwanted operations like moving constituents covertly to the relevant projection even if the focused or topical element stays in situ. Furthermore, it is a good exemplification of minimalism in general, which dictates "to examine every device [...] to determine to what extent it can be eliminated in favor of a principled account [...], going beyond explanatory adequacy" (Chomsky 2004b: 106). It is this 'going beyond explanatory adequacy' that can also be applied to derivational complexity, i.e. to the structure building process itself, as we show in the following section.

7.4 Derivational complexity, simplest Merge, and recursive layering

Recall from section 7.2 that early discussion of the complexity of a generative grammar (e.g. Chomsky 1956, 1957) established that the derivation of natural language recursive, self-embedding structures requires a formal system with the complexity of (at least) a context-free grammar. We see two developments in current linguistic minimalism that give rise to a new evaluation of the outcome of that discussion.

First, linguistic minimalism is characterized by a more articulated model of the grammar, where what takes place in the syntactic component ('narrow syntax') is reduced to the elementary process of combining elements ('Merge'). Other processes formerly considered to be part of syntax have been relegated to the interface components (including such hallmarks of transformational analysis as head movement and ellipsis, e.g. Chomsky 1995: 229). This new division of labor among the component parts of the grammar raises new issues as far as complexity measurement is concerned. In particular, the question of the complexity of grammar has to be answered separately for the grammar as a whole (narrow syntax, the interface components, and the interactions among them) and for the core component of narrow syntax, with different answers forthcoming in each case.

Second, linguistic minimalism is characterized by a more articulated concept of the derivation, i.e. the sequence of operations Merge turning a set of elements ('Numeration') into a hierarchical structure. In current minimalism, as mentioned above in section 7.3.1, such a derivation is taken to be punctuated, consisting of various phases (Chomsky 2001) or derivation layers (Zwart 2009a), each feeding the interface components independently of other phases/layers that are part of the same derivation. As argued in more detail below, this punctuated nature of the derivation calls for a reconsideration of what it is that makes a derivation recursive, thereby bearing on Chomsky's original argument for natural language grammars as being of the context-free type.

Let us begin the discussion of the complexity of grammar in linguistic minimalism with the hypothesis that narrow syntax is a maximally simple system containing only the structure building operation Merge. Merge selects two elements and combines them, yielding a set (Chomsky 2001: 3). Hierarchical structure is derived because the newly formed set is among the elements that Merge can select for its next operation. Since the output of one Merge operation can be the input for the next, Merge is taken to yield the property of recursion (we will modify this conception of recursion below). To facilitate a comparison between Merge and the rewrite rules of early generative grammar, it is necessary to consider the process from a top-down perspective, with the newly created set as the start symbol for that particular rewrite rule (to the left of the arrow), to be rewritten as the pair of elements merged. From the nature and number of those elements (terminal or nonterminal), it is possible to determine the complexity of the structure building process. Restricting ourselves to finite-state and context-free grammars, the type of rewrite rules to consider are:[7]

(10) a. finite-state: $A \rightarrow a \mid a \; B$
b. context-free: $A \rightarrow (a)^{*} \; (B)^{*}$

[7] In (10), ordinary characters are terminals and capitals nonterminals, | is the disjunction symbol, an* indicates an arbitrary number, and order is irrelevant.

The crucial difference between (10a) and (10b) appears to be the restrictions in terms of the number and nature of the elements yielded by the rewrite rules: in a finite-state grammar, the number of elements is at most two, of which at most one is a nonterminal, while in a context-free grammar, any number of terminals and nonterminals can appear. Note that we are not considering the *empirical* adequacy of each type of rewrite rule here, rather just the question of which type best captures the operation Merge, as currently understood. It would seem that the finite-state type (10a) comes closest, as the context-free type (10b) does not reflect a crucial property of Merge, namely that it combines two and only two elements. This suggests that the question of the complexity of the grammar, when applied to the component of narrow syntax, potentially yields a different answer ('finite-state') from the complexity question applied to the grammar as a whole, which includes narrow syntax, plus the interfaces, and the interaction among these components.[8]

There are, nonetheless, a couple of discrepancies between Merge and the finite-state rule type (10a). First, nothing in the rule seems to exclude the possibility that Merge can combine two nonterminals. Second, the finite-state rule may yield a single terminal, but Merge (by definition) cannot combine a single element (as Merge is defined as a process combining two elements). Perhaps both discrepancies disappear if a sister pair consisting of two phrases is disallowed because of labeling problems (Chomsky 2013) and if 'self-Merge' is what starts the derivation from a single terminal (Adger 2013: 19). However, it seems to us that there is a more natural way in which both discrepancies can be shown to be illusory.

First, we may simply assume that the first Merge operation in a derivation involves 'zero' as one of its terms (Zwart 2004; Fortuny 2008; De Belder and Van Craenenbroeck 2011), so that (10a) should be read as producing a terminal and either a non-terminal or nothing. Merge can produce a terminal and either a non-terminal or nothing, if we take Merge to be an operation that takes an element from some resource and adds it to the object under construction. The element taken from the resource is by definition a terminal, and the object under construction is typically more complex, being the result of earlier steps. This essentially iterative procedure yields binary branching, but since the object under construction is empty at the first step, the output of the first operation will be just a single terminal (Zwart 2004). This conception of Merge covers both parts of the disjunction in (10a), the rewrite rule yielding a terminal at the first step of the procedure, and a pair consisting of a terminal and a nonterminal after that. Similarly, in the top-down procedure proposed in Zwart (2009a), the final step in the procedure will involve just a single terminal, leaving only the empty set as what remains of the Numeration. Since the

[8] If so, sentences like (2a–b), which Chomsky argued show that natural language grammar is not context-free, should be reconsidered, for which see Zwart (2014) and below.

Numeration, in this system, is a nonterminal, each step yields a pair of a terminal and a nonterminal, except the last, which yields just a terminal (or a terminal and an empty set).

The discussion in the previous paragraph shows that slight modifications in the conception of Merge yield the result that Merge is essentially a finite-state type rewrite rule. Importantly, these modifications should not be seen as complications: whereas the standard conception of Merge stipulates that the number of elements manipulated should be two, these modified conceptions involve rules that manipulate just a single element at each step of the procedure (either adding it to the object under construction or splitting it off from the Numeration).

Second, the restriction that Merge, if it is to be of the type in (10a), never yields a pair of nonterminals, follows immediately once we understand that 'terminal' and 'nonterminal' should be defined relative to a derivation. This is where the punctuated nature of the derivation, mentioned above, becomes relevant. Assume as before that every derivation starts from a predetermined set of elements (the Numeration). Then a derivation maps the Numeration to a hierarchical structure. A terminal, then, is an element from the Numeration, whereas a nonterminal is essentially a stage in the derivation (e.g. the output of a step in the derivation). Since the Numeration may contain a phrase (construed in a previous derivation), a distinction needs to be made between the concepts of terminal/nonterminal on the one hand, and head/phrase on the other: A terminal can be either a head (X^0) or a phrase (XP). The finite-state rule (10a), then, while stating that a pair of nonterminals cannot be generated, does not state that a pair of phrases cannot be generated. What it does state is that one of the elements generated must be taken from the Numeration, its phrase structure status being irrelevant. Conversely, if an element of the Numeration is a phrase (the output of a separate derivation), merging it to the object under construction (yielding a pair of phrases) can be barred only by stipulation, and therefore should not be disallowed.

That derivations can be layered, with one subderivation feeding another, need not be stipulated. Indeed, it seems to be a feature of each grammatical component. For example, constructions manifesting derivational morphology, including compounds, are generated in separate, self-contained derivations, whose outputs may enter as atomic items in a syntactic derivation. Hence there is no reason to believe that this cyclic organization of the derivation should stop at the arbitrary boundary of 'words,' or that we could not have separate subderivations inside the syntactic component, one subderivation generating a phrase or a clause which shows up as a terminal in another. Essentially, this is the system of generalized transformations of early generative grammar (cf. Chomsky 1975 [1955]: 518). Arguably, the notion of *construction* as entertained in Construction Grammar (Goldberg 1995; Croft 2007) could be incorporated in the minimalist framework if we take constructions to invariably be the output of such a subderivation (Zwart 2009c).[9] Thus, the layered-derivation architecture contemplated here, to our mind, shares basic assumptions with the 'combinatorial interface rules' proposed by Jackendoff and Wittenberg (this volume, chapter 4). In particular, since Jackendoff and Wittenberg argue for uniform combinatorial operations regardless of "whether the constituent C is an utterance, a phrase, or a word, and whether its parts are phrases, words, or sub-word morphemes," their model shares our basic assumption that the opposition between words and phrases is artificial in the context of deriving syntactic objects.

The layered-derivation analysis might in fact be forced upon us by considerations of derivational simplicity. A grammar deriving a simple sentence like *The man left* from the Numeration in (11a) would not be able to proceed in a maximally simple fashion, merging a single element with each step of the derivation, as doing so would wrongly generate the constituent *man left* (illustrated for bottom-up unary Merge in (11b)). To get *the man* as a constituent in the output of the derivation, the Numeration would have to be as in (11c), where *the man* is the output of a separate derivation layer, and hence a phrasal terminal.

(11) a. { the, man, left }
b. step 1: merge *left* yielding **left**
step 2: merge *man* yielding **man left**
step 3: merge *the* yielding **the man left**
c. { [the man], left }

We refer to Zwart (2009a; 2011a) for a characterization of layered derivations and the idiosyncratic properties (including locality) that they yield. As stated above, accepting the possibility of layered derivations, terminals can be phrases, and the differences between Merge (in its simplest conception) and the finite-state rule in (10a) disappear.

We conclude, then, that Merge, as currently understood in linguistic minimalism, has the formal characteristics of a finite-state rewrite rule. It follows that if narrow syntax contains just Merge, a core component of the grammar may be characterized as having the minimal complexity of a finite-state grammar. Naturally, applying the question of complexity to the grammar as a whole yields a different answer. Nevertheless, we take the conclusion that narrow syntax is finite-state to be potentially important, especially if narrow syntax is to be equated with the faculty of language in the narrow sense, as argued by Hauser, Chomsky, and Fitch (2002).

[9] Not allowing elements in the Numeration to be phrases would increase the complexity of the grammar significantly, as the derivation would then have to involve subroutines, with the derivation feeding back into the Numeration (e.g. Bobaljik 1995) or involving parallel routes (e.g. Citko 2005). However, as pointed out in Zwart (2014), if the output of these subroutines or parallel routes showed idiosyncratic sound-meaning properties, the subroutines or parallel routes would have to be connected to the main derivation via the interfaces, and we would essentially be looking at layered derivations.

If there is any merit to our conclusion that narrow syntax is finite-state, we need to reconsider the argument from recursive self-embedding sentences of natural language, such as (2a–b), which it will be recalled was taken to prove that the complexity of grammar of natural languages has to be at least of the context-free type. We submit that the punctuated nature of the derivation, in particular its use of derivation layers, calls for a reconsideration of the question of how recursion is brought about.

In fact, a layered-derivation architecture is inherently recursive, as the output of one instance of a procedure *P* (a derivation layer) serves as input for another instance of *P* (Chomsky 1975 [1955]: 518; Zwart 2011b; Trotzke and Lahne 2011). A derivation as a whole, then, can be recursive even if its constituent subderivations are little finite-state grammars. The early discussions of the formal properties of natural language grammars were correct in stating that natural languages like English are not finite-state languages. However, it seems to us that this does not justify the conclusion that the rules of grammar (Merge in its simplest conception) are not finite-state. The complexity of the grammar as a whole is not a function of the nature of Merge, but of the recursive process of derivation layering. And the complexity of language is greatly increased by whatever takes place at the interfaces, which is irrelevant to the proper characterization of Merge. Accordingly, as for the complexity of the grammar as a whole, we acknowledge the reasoning behind the idea of "moving in the direction of construction-based grammar, which makes the 'interface' the heart of the entire grammar" (Goldberg 1996: 14).

Our discussion leads to the conclusion that the complexity of the grammar needed to derive recursive self-embedding sentences follows automatically from the punctuated nature of the derivation, i.e. from the assumption (inevitable, if we are correct) that derivations can be layered complexes of maximally simple subderivations. The arguments against grammar being finite-state, based on sentences like (2) and (3), then, are relevant to the complexity of the grammar as a whole, that is, narrow syntax in conjunction with the interface components and, crucially, the interaction among derivation layers, but not to the complexity of narrow syntax itself. It would take us too far afield to discuss the derivation of sentences like (2) here in any detail. However, a crucial property of these constructions may be pointed out, namely that they inevitably involve clausal embedding. If clausal embedding involves layered derivations (i.e. the interaction of subderivations, essentially in the form of generalized transformations), then constructions of the type in (2) are relevant to the question of the complexity of the grammar as a whole, not to the question of the complexity of each subderivation (i.e. each individual sequence of Merge operations). As before, the conclusion that English is not a finite-state language does not entail that narrow syntax is not finite-state.

7.5 Conclusion

In this chapter, we have discussed minimalist notions of formal complexity measurement and argued that both the derivational and the representational consequences of linguistic minimalism result in an approach that shares basic assumptions with alternative perspectives represented in this volume. After pointing out the distinction between complexity measurement at the level of performance and at the level of grammar in section 7.2, we sketched core minimalist ideas of complexity measurement. In section 7.3, given this general background, we showed that reduction in the domain of representational complexity results in points of convergence between minimalism and a "methodology of assuming as little syntactic structure as necessary" (Jackendoff and Wittenberg, this volume, chapter 4). However, while proponents of Simpler Syntax often call their approach a 'different sort of minimalism' by referring to the amount of representational levels and of covert operations in mainstream generative grammar, we showed, based on our analysis of one specific phenomenon from German, that complexity reduction in the sense of Simpler Syntax follows from the standard minimalist methodology. After discussing representational complexity in the context of left-periphery-movement in German, we turned to the issue of derivational complexity from the perspective of current linguistic minimalism. We argued that, as far as overall complexity is concerned, narrow syntax must be distinguished from the interface components, and one must keep in mind the punctuated nature of the derivation, which involves phases and/or derivation layering. In sum, narrow syntax can be captured by a finite-state device and, therefore, falls low on the Chomsky hierarchy. Furthermore, given that not only words, but also phrases and clauses can be atomic items in the course of a syntactic derivation, we concluded that the notion of 'construction,' as entertained in alternative frameworks, can be fruitfully implemented in a minimalist system.