Symmetry Breaking in Syntax and the Lexicon

Linguistik Aktuell/Linguistics Today (LA)

Linguistik Aktuell/Linguistics Today (LA) provides a platform for original monograph studies into synchronic and diachronic linguistics. Studies in LA confront empirical and theoretical problems as these are currently discussed in syntax, semantics, morphology, phonology, and systematic pragmatics with the aim to establish robust empirical generalizations within a universalistic perspective.

For an overview of all books published in this series, please see
http://benjamins.com/catalog/la

Volume 216

Symmetry Breaking in Syntax and the Lexicon
by Leah S. Bauke

Symmetry Breaking in Syntax and the Lexicon

Leah S. Bauke
Goethe University Frankfurt

John Benjamins Publishing Company
Amsterdam / Philadelphia

The paper used in this publication meets the minimum requirements of the American National Standard for Information Sciences – Permanence of Paper for Printed Library Materials, ANSI Z39.48-1984.

Library of Congress Cataloging-in-Publication Data

Bauke, Leah S.

Symmetry Breaking in Syntax and the Lexicon / Leah S. Bauke.

p. cm. (Linguistik Aktuell/Linguistics Today, ISSN 0166-0829 ; v. 216)

Includes bibliographical references and index.

1. Grammar, Comparative and general--Syntax. 2. Minimalist theory (Linguistics) 3. Lexical-functional grammar. 4. English language--Syntax. 5. German language--Syntax. 6. Russian language--Syntax. 7. English language--Grammar, Comparative--German. 8. English language--Grammar, Comparative--Russian. I. Title.

P158.28.B38 2014

415--dc23 2014011671

ISBN 978 90 272 5599 0 (Hb ; alk. paper)

ISBN 978 90 272 7012 2 (Eb)

John Benjamins Publishing Co. · P.O. Box 36224 · 1020 ME Amsterdam · The Netherlands
John Benjamins North America · P.O. Box 27519 · Philadelphia PA 19118-0519 · USA

For my nan
- in memoriam -

Table of contents

Acknowledgements

This book would not have been possible without the help and support of my colleagues, friends and family. I would like to express my sincere thanks to all of them, all of it is greatly appreciated.

List of abbreviations

ACH	abstract clitic hypothesis
ACP	abstract clitic position
AS	argument structure
BPS	bare phrase structure
CED	condition on extraction domain
C-I	conceptual-intentional domain
DM	distributive morphology
ECM	exceptional case marking
ECP	empty category principle
EF	edge feature
EM	external Merge
EPP	extended projection principle
FI	full interpretation
FL	faculty of language
FLN	faculty of language (narrow sense)
IC	inclusiveness condition
IM	internal Merge
ING	incorporated nominal gerund
LCA	linear correspondence axiom
LF	logical form
LE	linking element
LI	lexical item
MLC	minimal link condition
NTC	no-tampering condition
PF	phonetic form
PHON	external system for sound
PIC	phase impenetrability condition
PING	pluralized incorporated nominal gerund
PoS	point of symmetry
P&P	Principles and Parameters (Theory)
RCP	root compounding parameter
SC	small clause
SCHM	successive cyclic head-movement
SEM	external system for meaning
S-H-C	specifier-head-complement order
SIMPL	simplify operation
S-M	sensory-motor domain
SMT	strongest minimalist thesis
SO	syntactic object
*u*F	unvalued feature
uFF	uninterpretable feature

CHAPTER 1

Introduction

1.1 Background

Deeply routed in the generative tradition, as outlined in Chomsky's writings from early on, (cf. e.g. Chomsky 1955; 1957; 1959; 1964; 1965; 1966; etc.), in recent years Minimalism has developed more and more as the logical next step and as a perfectly reasonable and promising idea to entertain. This does not mean that a minimalist agenda is not exposed to criticism. In fact even its most fundamental underlying assumptions are questioned (cf. also e.g. Hornstein 2009: 1). These fundamental assumptions, which Boeckx (2006: 88–91) identifies as the core of a minimalist research agenda that needs to be protected and defended against all obstacles, have not changed (much) since they were first laid out clearly and extensively in Principles and Parameters (P&P) terms (cf. e.g. Chomsky 1981; 1982; 1986a; 1986b; 1991 and related literature). This alone should suffice to dismantle unsubstantiated criticism and make the minimalist core more than just a scientific guess or belief (cf. Boeckx 2006: 88). Additionally, research in language acquisition produces results that point into the direction of a minimalist theory of language (cf. e.g Roeper 2007; forthcoming) and eventually, research in related disciplines such as evolutionary biology and cognitive psychology (cf. e.g. Hauser, Chomsky & Fitch 2002) further substantiates the idea that minimalism is the closest shot that has been aimed at the target of a biolinguistic approach so far.

Still, this does not imply that all of the criticism directed against a[1] minimalist account is unmotivated. Even among supporters of minimalist thinking criticism is voiced. Some of it is procedural and directed against very specific technical implications, some of it is more fundamental, calling into question the overall validity of certain basic assumptions that might even have been regarded as unequivocally sound and proven. Both of this criticism is important and necessary for further advancing the research agenda. Some of it arises naturally from the programmatic nature of the minimalist program (cf. e.g. Boeckx 2006: 85–87) that leaves plenty of room for 'alternative instantiations' (ibid: 5; cf. also Hornstein 2009: 4 for a similar point), which eventually might turn out to be contradictory in some aspects. One of these possible instantiations is put forward in the claim that

1. I use the indefinite article here in Chomsky's sense.

Minimalism is best defined in terms of Occam's razor (cf. e.g Radford 1997: 515). While the role of the razor is hardly ever denied by practitioners of minimalist approaches (cf. e.g. Hornstein 2001: 14; 2009: 2), many hasten to point out that it does not qualify as the defining characteristic of minimalism. Boeckx (2006: 7) describes minimalism in Occam's razor terms as a farce,[2] Gallego (2010: 2) makes the important distinction between methodological minimalism and ontological[3] minimalism and argues that the latter crucially 'defines the M[inimalist] P[rogram] as something brand new' that sets it apart from P&P accounts. So, ontological minimalism rather than methodological minimalism is the standard of comparison for the qualification of technical solutions, which, due to the aforementioned programmatic character of the MP are often offered in numerous different and almost indiscriminate guises.[4]

The question that immediately emerges in this context is what the standard of comparison, or what ontological minimalism then is. The answer is that ontological minimalism provides answers to the question how the faculty of language (FL), i.e. the human mind, interacts with other biological systems (cf. Gallego 2010: 2). This includes the question of how much is unique to FL and how much can be accounted for in what Chomsky (2004; 2005; 2008) characterizes as third factor terms and what is needed for optimal interaction between FL and other human systems. The strongest possible assumption, which, as Chomsky himself acknowledges, is probably too strong, is expressed in the Strongest Minimalist Thesis (Chomsky 2000: 96):

(1) Strongest Minimalist Thesis (SMT):
    Language is an optimal solution to legibility conditions.

Nevertheless, it is common practice and quite reasonable to start with the strongest possible assumption in scientific investigations of any kind (cf. Boeckx 2006: 4) and then carefully determine where deviations from this ideal path are necessary. This, as Chomsky (2008: 135) points out, is what in the context of linguistic

2. It is also worth noticing that the dualism between global and local economy can be seen as a result of reducing minimalism to methodological aspects only (cf. e.g. Boeckx 2006: 100–101).

3. Gisa Rauh p.c. remarks that ontological minimalism is also referred to as empirical minimalism in the literature. I treat both terms on *a par* here and will continue using Gallego's term of ontological minimalism.

4. Tom Roeper p.c. once pointed out to me that when there are too many possible solutions, this is a good indication that something is wrong with your theory, (*viz*. your standard of comparison). A conceptually and theoretically sound argument should allow for only one possible solution. This is immediately evident also from the perspective of language acquisition. It is quite unlikely that – all things being equal – FL allows for more than one option from which the learner has to chose, as this would spuriously delay the learning process.

research leads to a better understanding of how FL evolved and thus provides answers in line with ontological minimalism. To be clear here, Minimalism, which is shaped by methodological **and** ontological aspects, then does not aim at providing an even more accurate description of the data than that provided e.g. by the P&P theory. If a more precise description (and explanation) of the data happens to fall out from minimalist investigations, this, of course, is a welcome result, but it is a mere side-effect that follows from a more accurate and precise analysis of FL itself (cf. Boeckx & Uriagereka 2007; Gallego 2010: 4).

In line with this tradition, this study aims at both: sharpening the understanding of FL and its interaction with other systems of the human mind and providing for a better and more precise description of heretofore unaccounted empirical data. So all analysis of empirical data is subjected to careful scrutiny and evaluated against the SMT. The data investigated here are nominal root compounds, incorporated and non-incorporated nominal gerunds and small clauses and it is shown that the analyses suggested in this context carry over to related domains such as light verb constructions, verb-particle constructions and subject auxiliary inversion. As expected from research carried out pledging to the standards described above, it turns out that analysis of the data that is guided by the SMT does not only qualify as explanatorily more adequate than other accounts but also provides invaluable insights for the conception of minimalist theory.

However, before engaging into detailed analyses a few more thoughts on the SMT are in order. The SMT states that language is an optimal solution imposed on FL by interface requirements (cf. also Chomsky 2008: 135). The two interfaces that Chomsky identifies as a virtual necessity are the C-I and the S-M interface (cf. Chomsky: 1995a: 168–169). Hence, in the ideal case all demands that arise from the interaction of FL and these two interfaces can be satisfied in line with the SMT. This is probably too strong a statement, but all deviation from this assumption 'merits close examination, to see if it is really justified' (Chomsky 2008: 135). Additionally, it needs to be determined whether any one of the two interfaces has prominence over the other or whether interaction between FL and C-I and S-M is *pari passu* without any superseding effects. Again, Chomsky points out that language has traditionally been identified as more fit to cope with LF-demands than with PF-demands. This has been the tenor in the study of language carried out in the longstanding humanist tradition of von Humboldt and is corroborated by research in biology most recently (cf. Chomsky 2008: 136; Hauser, Chomsky & Fitch 2002; Fitch, Hauser & Chomsky 2005: 200). If true, which pending qualified scientific research in biology is almost impossible to verify but at least a reasonable assumption, various conclusions can be drawn here. Either, LF actually takes prominence over PF and PF, being the interface that feeds ancillary processes, can be treated as just that, i.e. the place to deal with ancillaries. However, although not totally out of the way, this stands in stark contrast to what minimalist research in the past decade has dealt with (cf. Chomsky 2008: 135). Alternatively, it could be argued that LF and PF do not interface with FL at the same time. A solution that *a priori* does not provide much of an insight on whether one of the interfaces supersedes the other and that is introduced in Chomsky (2000) and controversially discussed in great detail in Epstein & Seely (2002); Sauerland & Elbourne (2002); Marušič (2005); Bošković (2007); Hicks (2009) and others. The still other alternative is that LF might actually be more fit to deal with the requirements of language but that the role of PF-interactions should not be diminished by that. This is the stand taken in this study, which inspects all analyses of empirical data with respect to its LF- and to its PF-compatibility in accordance with the SMT, which, it is important to notice, is formulated in a way that does not give precedence to any of the two virtually conceptually necessary interfaces. Now, it is time to turn to the question of what it is that FL generates that, in the ideal case, feeds the interfaces systems of PF and LF. In plain terms, this is a hierarchically ordered, recursively embedding, monotonic syntactic structure (cf. among many others Fitch, Hauser & Chomsky 2005; Chomsky 2005: 7–9; Hornstein 2009: 53ff.; Gallego 2010: 30–31), the derivation of which is discussed in the next chapter.

1.2 Merge (concatenation and labeling)

Merge[5] is characterized as the one operation that applies freely in narrow syntax (cf. e.g. Chomsky 2004: 117). As such, this cost-free operation is 'most fundamental' (Gallego 2010: 139) and very basic (cf. e.g. Boeckx 2008a: 79) at the same time. It is an operation that applies to lexical items[6] (LIs) and not to features. So, what Merge does is that it takes an LI from the Numeration or straight from the lexicon, depending on which conception proves more viable,[7] and combines this either with another LI taken from the lexicon or the Numeration thus creating a syntactic

5. Merge, in this first approximation here, is to be understood as Set-Merge. The distinction between Set-Merge and Pair-Merge will be included in the discussion at a later stage.

6. Gallego (2010: 6) makes this important distinction and points out that lexical items are to be understood in this context in a non-distributed sense. For the time being this is also how the term is to be understood here. However, the discussion in Chapter 2 will provide a more detailed investigation into the question what items Merge can operate on.

7. The question whether LIs are selected from the Numeration or straight from the lexicon is an important one and is discussed controversly among others in Frampton & Gutmann (2002); Stroik (2009); Grewendorf & Kremers (2009); Richards (2010) and Gallego (2010). Since it does not have a direct impact on the discussion here, however, I remain agnostic to it for the most part (but cf. Chapter 5 for some elaboration).

object (SO), or it combines an LI selected from the Numeration or the lexicon with an already existing SO, also resulting in a SO. Hence, in any case the output of Merge is a SO. Those LIs that are accessible to Merge, i.e. those that have the property of being potential Merge candidates, bear a feature signaling this property. Chomsky (cf. 2008: 139) identifies this feature as an edge feature (EF).[8] In line with the SMT, this is all that is needed to create SOs that are handed to the interfaces for interpretation. As Chomsky notes, if on the right track, tying the applicability of Merge to EFs has the effect of not only rendering d-structure and s-structure as unformulable, but also eliminating mapping operations, which were needed for mapping d-structure onto s-structure and eventually onto LF in earlier approaches; so ultimately this conception of Merge provides for a purely minimalist account of what used to be cast in terms of generalized transformations in very early approaches (cf. ibid). Merge then is a structure building operation – in fact, in line with the SMT it is even more appropriate to characterize Merge as **the** structure building operation – that applies recursively in narrow syntax and that builds syntactic structures. Apparently, the only thing needed to set off the iterative operation, are LIs with EFs. Chomsky points out further, that EFs do not form part of a Match-Valuation process (cf. ibid: 2007; 2008). This means that, unlike other features, EFs do not enter the derivation as unvalued features and will never be valued in the course of the derivation. Further, this implies that Merge, when triggered by EFs alone, does not pay any attention to further restrictions coded in featural terms (cf. e.g. Gallego 2010: 7). Features do not require features (cf. Adger 2010; Boeckx 2010a: 17). Merge so defined is potentially unbounded and thus yields structures of discrete infinity, which, as Chomsky (2008: 138–139) never gets tired of pointing out, can be regarded as the defining property of FL that lead to the great leap forward in evolutionary terms (cf. also Hornstein 2009 for a detailed but relatively brief overview). However, despite the fact that Merge defined in terms of EFs does not violate the SMT, a number of questions remain unanswered at this point.

For one, it is not at all clear, what initiates the unbounded operation. For the very first Merge operation only one of the two possibilities described above is at hand. A LI taken from the lexicon/Numeration is merged with another LI also taken from the lexicon/Numeration. Both of these LIs bear EFs that are active, because EFs – unlike other features – are always active an never delete (cf. Chomsky 2008; Gallego 2010; Narita 2009; 2011). Concentrating on EFs may thus suffice to set off a first Merge operation, but it is not clear how the derivation proceeds from this point onwards and in retrospect it is also not clear how the derivation got

8. Note that this definition of EF differs significantly from that used in previous analyses in which EFs were basically equated with EPP features on Phase-heads (cf. Chomsky 2008: 154–155 and Gallego 2010: 45–46 for valuable discussion).

to this point in the first place. In order to illustrate this more clearly, the output of Merge and the character of the operation itself needs to be taken into account.

The output of Merge is, in any case, a SO not a LI and the operation itself is source independent[9] and order independent (cf. Boeckx 2008a). Hence, in principle any two LIs, with EFs say α_{EF} and β_{EF}, can be merged in any random order so (2a) and (2b) can be regarded simply as notational variants:

(2) a. b.

The input to Merge in this case are two atoms, which – by virtue of the operation applying to them – are combined to a set $\{\alpha, \beta\}$. This goes to show that the operation itself is completely symmetric and can just as well be described as concatenation (cf. Boeckx 2008a: 79–80; Hornstein 2009: 54–55; Pietroski & Hornstein 2009).[10] However, it is widely believed that the output of Merge is not symmetric (cf. e.g. Boeckx 2008a: 79–81; Gallego 2010: 7–9; Chomsky 2008: 145). Hence, in order to introduce asymmetry into the structure arguably a label needs to be determined (cf. also Donati & Cechetto 2010: 1). Determination of a label merits two positive side-effects. First of all, it introduces hierarchical order into the structure, i.e. it does not simply produce a set but an ordered one at that, and secondly, it constrains Merge in such a way that what is described as wild-type Merge (cf. Boeckx 2008b: 3) or chaotic Merge (cf. Gallego 2010: 7) is given far less room than it seems to be granted at first sight. Hence, taking the order independence of the Merge operation seriously, the output of any of the Merge operations described in (2) could be something like either (3a) or (3b):[11]

9. The source independence of Merge is still hotly debated and it is far from clear whether External Merge (EM), i.e. Merge of an LI from the lexicon/Numeration with another LI from the lexicon/Numeration or with an already existing SO, is really on *a par* with Internal Merge (IM), i.e. Move of an already merged LI/SO to another position (cf. e.g. Hornstein 2009: 128–29 vs. Chomsky 2008: 140–141). This question will be dealt with in detail later on. For now Merge is to be understood as EM only.

10. This is an assumption that is not generally agreed upon. DiSciullo (2005) and DiSciullo & Isac (2008a;b), for instance, argue that Merge is inherently asymmetric. This is an interesting assumption, however, it comes at the cost of restricting Merge far more narrowly than originally described in SMT terms. How a useful restriction of Merge, that is definitely needed (cf. below), could look like will be discussed in due course.

11. Actually, concluding that the asymmetry is introduced by making either α or β the relevant label already presupposes that the label is determined from one of the Merge partners (as required by the Inclusiveness Condition of Chomsky (1995a: 228)) and that it is determined neither by intersection nor by union but rather by identity (cf. ibid 244) – but cf. Citko (2008a;b) for arguments in favor of union.

(3) a. b.

{β{α, β}} {α{β, α}}

This then means that while the operation of combining two LIs or one LI and an already existing SO is symmetric, the output of this operation is asymmetric, i.e. an ordered set. This, of course begs the question, how this asymmetry is introduced or to be more precise how the label is determined.

One possible solution is to simply determine complements and specifiers in terms of first Merge and later Merge, as suggested in Chomsky (2005: 14). Although this is perfectly viable, it requires the derivation to keep track of the order in which elements were merged, which is an assumption that is hard to maintain unless motivated independently (cf. e.g. Boeckx 2008b for an alternative suggestion going into that direction). Two other solutions discussed in the literature (cf. e.g. Guimarães 2000; Fortuny 2006) are to assume that a LI selected from the Numeration/lexicon either merges with an empty set, which automatically makes the LI the label-providing element, or it merges with itself, which again leads to the fact that the LI provides the label (although it obscures which of the two Merge partners is the label producing one and leads to the question of how the two instances of the same LI can be justified (cf. also Boeckx 2008a: 80–81 for further criticism)).

Still another possibility is to let the interfaces determine the label of the structure. This is basically what is suggested in Moro (2000; 2007). Here Merge again resembles Concatenation in so far that it produces symmetric structures, and asymmetry is introduced via a movement operation that moves one of the Merge partners to a higher specifier position. This produces a structure in which the moved element asymmetrically c-commands the unmoved Merge partner and by virtue of that produces a structure that is in line with the Linear Correspondence Axion (LCA) (cf. Kayne 1994). The LCA is a mechanism that maps asymmetric c-command to linear precedence and thus provides a function for linearizing hierarchical structure at the PF interface. [12]

Although this again is a perfectly sound description of how labels could be determined, this account just rephrases the initial question of label-determination. Instead of asking which of the two Merge partners produces the label, the question is

12. While Kayne (1994) is at great pains to illustrate that the LCA does not apply at PF only, this is the place where Chomsky (1995a) relegates it to. Moro (2000) is not very explicit about whether the LCA applies at PF or in narrow syntax, but it is pretty clear that the movement operations that Moro postulates are performed against considerations on PF-compatibility.

now shifted to which of the two Merge partners is the one that moves. Moreover, this conception of Merge ultimately endorses an account in which the interfaces, and here in particular the PF-interface are equipped with their own computational capacities which, as Boeckx (2008a: 84) argues, is an undesirable result. This result is undesired conceptually, for the reasons described above and it is undesired technically, because this is what brings back in the question of how the derivation is supposed to proceed after the first Merge operation took place. Further Merge is not licit as long as it cannot be determined what a new Merge partner is merged to (cf. Boeckx: ibid) and this can here only be determined upon contact with the PF-interface. This then would suggest that every Merge operation requires immediate Transfer at least to the PF-interface. Although approaches that require immediate Transfer exist (cf. e.g. Epstein & Seely 2002), this is hardly ever Transfer to PF.

In any case, such an analysis brings up the question of LF-Transfer. If asymmetry is determined at PF in order to guarantee that the syntactic structure is LCA compliant and can thus be linearized straightforwardly at the PF-interface, then how is the structure interpreted at the LF-interface? One answer could be that the requirements of PF are somehow already taken care of in narrow syntax (cf. also footnote 12). This avoids most of the complications mentioned above but brings up two other questions. The first one is how FL 'knows' what the requirements that will be imposed by PF are and the second one is why of all it is PF that somehow guides the computation when this is arguably only the place for ancillary processes (cf. Chomsky 2008: 136).

So another option is to let LF-requirements do the label-determination. Chomsky entertains this idea in his writings for some time (cf. e.g. 2000; 2001; 2004). Here it is s-selectional requirements, which establish some sort of a 'is-a' relation already prominent in Chomsky (1955), that determine which of the Merge partners produces the label. This, however, is problematic from various perspectives and in various respects. If s-selection is post-syntactic in the same way that linearization is on the PF-side, this conception is just as problematic and for the very same reasons. If s-selection is part of narrow syntax, this is an unwelcome deviation from core minimalist assumptions, as Chomsky himself notes (cf. 2004: 112). This then means that s-selection is not a solution to the question of label-determination regardless of whether it might apply in narrow syntax or at LF. [13]

Eventually, any approach that gives prominence to or identifies any one of the two interfaces as the driving force for label-determination runs into the trouble of

---

13. Boeckx (2008a: 91–93) shortly discusses s-selection and also c-selection as suggested in Collins (2002) (cf. also Stabler (1998) and Müller (2010) for alternative accounts on c-selection) in narrow syntax as a possible solution, but ultimately discards it in favor of f-selection, i.e. selection on the basis of unvalued φ-features, as the label-determining characteristic, as discussed below.

not being able to explain why labeling plays a role at **both** interfaces in the same relevant way, i.e. either for linearization or for monadic predicate formation (cf. Boeckx 2008a: 66–71; Moro 2000; Pietroski 2002; 2006). This strongly implies that label determination takes place in narrow syntax and the only remaining question then is how this is done. Here three alternatives that are currently floating in the literature are looked at and it will turn out that neither one of them *prima facie* fares well with the structures discussed in this manuscript. The first one is advocated in Gallego (2010: 13–15) and is closely related to Chomsky's (2008: 145) labeling algorithm:[14]

(4) In {H, α}, H is a lexical item (LI), H is the label.

This simply means that when a head is merged with an already existing SO, this head is automatically the projecting element. While this might be successful in most cases, it does not provide an answer to the First-Merge problem where arguably two heads are merged (cf. ibid: 12) and, as Cechetto & Donati (2010: 243–245) point out, is also potentially problematic in a number of other cases. With respect to First-Merge Gallego (2010: 13) argues, reminiscent of Boeckx (2009: 47), that this can be accounted for provided that Merge can produce unlabeled structures so long as it does not interact with the interfaces. This clearly dissociates Merge from labeling and allows for the emergence of unlabeled structures in narrow syntax. However, it allows for wild-type Merge right until the structure is sent to the interfaces for the first time, where deviant interpretations are filtered out (cf. Boeckx 2008b for a similar proposal). Again, this is a conception that provides a technically sound solution, but conceptually it raises a number of questions. First of all, it just so stops short of granting the interfaces some computational capacity of their own, which might turn out to be a welcome result given the SMT. However, it brings in the additional questions of when Transfer to the interfaces takes place, which in and of itself is a hotly debated issue (cf. e.g. Uriagereka 1999; Marantz 2001; Epstein & Seely 2002; Bošković 2005; Matushansky 2005; Svenonius 2004; Samuels 2008; Chomsky 2008 and many more) and of what exactly happens at the interface, which is by no means less controversial (cf. e.g. Chomsky 2004; 2008; Grewendorf & Kremers 2009; Gallego 2010). Economy considerations surely suggest that contact with the interfaces is made relatively early in the derivation. But, unless subscribing to a highly derivational approach in which every single Merge

14. Actually, Chosmky (2008: 11) introduces a second labeling algorithm.

(i) If α is internally merged to β, forming {α, β} then the label of β is the label of {α, β}.

Since this algorithm is concerned with Internal Merge (IM) and therefore not relevant to the problem of First-Merge, we can afford to ignore it at this point and will come back to it later in the discussion (cf. also footnote 17).

operation is sent to the interfaces straight away, as argued for in Epstein (et al.) (1998); Epstein (1999); and Epstein & Seely (2002), this leads to a characterization of FL in which the computational system is burdened with a considerable amount of parallel processing right from the start and even if the interfaces are contacted immediately after the first Merge operation, it is still the business of the interfaces to weed out deviant structures that were produced in parallel.[15]

A possible solution to this problem seems to come from Chomsky (2008), where he argues for what seems to be an even more refined version of the labeling algorithm described in (4) above. Particularly in cases of external Merge, label-determination is characterized as a dual operation:

> The label selects and is selected in E[xternal]M[erge], and is the *probe* that seeks a *goal* for operations internal to the SO: Agree or I[internal]M[erge]. (Chomsky 2008: 141)

This is a conception of Merge that ties labeling to the application of further operations such as IM or Agree. What is important is that these latter two operations apply within the SO formed by EM. This strongly suggests that the label must be determined prior to Spell-Out, because otherwise the operations IM and Agree were operative across Phase-boundaries in violation of both versions of the Phase Impenetrability Condition (PIC) (cf. Chomsky 2000: 108 vs. Chomsky 2001: 14; 2004).[16]

However, it should not be overlooked that, although this conception of Merge points into the direction that labeling takes place before Transfer to the interfaces, it comes at the price of integrating a selectional mechanism into the definition. This echos the LF-driven approaches to label determination discussed and discarded above. Thus, in order to approach Merge and label determination in such a way that the LF- and PF-independent character of the operation, as described in (4), is reflected and at the same time the additional refinements of Chomsky (2008: 141) are captured, Boeckx suggests a Probe-Label Correspondence Axiom (2008a: 96):[17]

(5) The label of {α, β} is whichever of α or β probes the other, where the Probe = Lexical Item whose *u*[nvalued]F[eature] gets valued.

15. To be clear here, this is not an argument for crash-proof syntax (cf. Frampton and Gutmann 2002) and against letting the interfaces decide on whether a structure built in syntax is deviant or not. It is simply an argument against letting syntax do parallel computations unless they are absolutely unavoidable.

16. Not to belabor the point, but if the label is what acts as a Probe for Agree or IM, then the label must be known prior to Spell-Out. Otherwise IM and Agree would operate across the Phase-boundary, which is something that goes directly against the assumption that Merge and thus also IM is always to the edge.

17. Cf. also Pesetsky & Torrego's (2006) Vehicle Requirement on Merge, which goes into the same direction.

Paired with the assumption that feature-Valuation never is a mutual process, this guarantees that a label can be determined unambiguously. Hence, even if two LIs (bearing some sort of *u*F each) are merged, only one of the *u*Fs probes and thus leads to feature-Valuation. The other LI may still value its *u*F under Agree, but this is dissociated from the process of Merge itself (cf. Boeckx 2008a: 96). In consequence, labels are determined prior to Spell-Out, parallel computations are avoided and neither LF- nor PF-properties are involved in the determination process. This last point is an important one. In particular the implementation of the distinction between valued and unvalued features (cf. Chomsky 2004), instead of the formerly advocated distinction between interpretable and uninterpretable features, alleviates FL from selectional restrictions associated with LF-requirements (cf. Gallego 2010: 25–27; Pesetsky & Torrego 2007).

Adger (2003: 91–92) already proposes a similar mechanism and Cecchetto & Donati (2010: 246) offer a labeling algorithm that seems virtually indistinguishable from the one in Boeckx:

(6) The label of a syntactic object {α, β} is the feature(s) that act(s) as a Probe of the merging operation creating {α, β}.

However, Cecchetto & Donati reject Boeckx's (2008a) assumption that in the course of the derivation 'only one relevant thing happens at any given time' (ibid: 96). Instead in Cecchetto & Donati's (2010: 246) system EFs count as Probes too. Hence, under First-Merge two LIs with EFs would both count as Probes and the LI bearing an additional feature then is a double Probe that wins over the single Probe and thus provides the label. Several things need to be taken into account in this context. First, on the one hand it seems only reasonable to include EFs into the consideration here, as these are, after all, the prerequisite for Merge. However, given that EFs never get valued and never delete (cf. Chomsky 2007; 2008), it is questionable whether they are of any relevance beyond the concatenative process of Merge and in particular whether they play a role in the process of label determination. In light of the fact that feature-Valuation is the crucial indicator for Probes and thus for label-providers in (5), integrating EFs seems to be an undesirable step. Second it is not at all clear how the derivation keeps track of single and double Probes, which would be essential, especially under a derivational approach to double Probes (cf. Cecchetto & Dontai 2010: 246). This is reminiscent of the problem of Merge keeping track of First- and Second-Merge operations by means of a counting mechanism, which is something FL seems to be unable to do (cf. Hornstein 2009: 37).

What gives rise to further skepticism in a 'double probing' approach is that probing is a two-step process. The first part is feature-Matching and only the second part is feature-Valuation (cf. Chomsky 2000; 2001; Gallego 2010: 27). Hence, in an algorithm that does not pay particular attention to feature-Valuation, as the one in (6), it is not clear what provides the label in cases of feature-Matching that is not followed by feature-Valuation.[18] Gallego (2010: 27–28) argues further that it is φ-features only that participate in the probing-process and Boeckx (2008a: 97) makes the same point in the context of the labeling axiom in (5). Cecchetto & Donati (2010: 245), however, argue that s-selectional features or categorial features[19] are likewise involved. The problematic status of s-selection has been discussed above already and will therefore not be reiterated here. With respect to categorial features Boeckx (2008a: 93) points out that these 'are notoriously stipulative in character [...] and very quickly dissolve into semantic criteria'. Therefore, the same skepticism that is directed towards s-selection in other accounts seems to be expedient here.[20]

This is also the reason why Hornstein's (2009) and Hornstein & Nunes' (2008) account on labeling in terms of Chomsky's (1955) 'is-a' relation is suspicious. However, nothing in these two approaches seems to stand in the way of recasting the 'is-a' relation that Hornstein (2009: 58–59) describes as a means of making the

18. Gallego (2010: 27–28) points out that what Chomsky (2000; 2001) identifies as defective core functional categories (i.e. *v* in contrast to **v* and T) are potential candidates for feature-Matching without feature-Valuation (cf. also the discussion in Chapter 5 for further elaboration).

19. Cechetto & Donati (2010: 245) are not clear about whether they make a principled distinction between c-selection and s-selection, but cf. Rauh (2010: 108–110) for a clear and concise distinction.

20. DiSciullo & Isac (2008a;b) advocate for an analysis in which Merge is inherently asymmetric. This asymmetry results from a proper inclusion relation that is operative on Merge. Hence, Merge can apply only to those LIs whose features are in a proper inclusion relation. However, a distinction is made between EM and IM operations (cf. DiSciullo & Isac 2008a: 270). For EM the label is determined on the basis of c-selectional features alone, while for IM the total set of features is taken into account. Yet, the total set of features explicitly excludes φ-features, which according to DiSciullo & Isac (i.e. 2008a: 269) 'are not computed in syntax, but in a different space'. Hence, the features looked at in IM cases are operator features, which occur only on Phase-heads. DiSciullo & Isac (2008a: 271–272) point out that this allows for predictions on when IM takes place: i.e. IM is a last resort operation that is available only in those cases in which EM is not an option simply because there is no LI left for EM in the subarray of the Numeration that is accessible in the current Phase. DiSciullo & Isac (2008a: 270) further argue that this makes the assumption that IM is invariably to the edge (cf. Chomsky 2008) follow naturally. This clearly predicts that movement of the subject to Spec, TP (cf. DiSciullo & Isac 2008a: 277) should be illicit contrary to fact and stands in stark contrast to several of the movement operations in DiSciullo & Isac (2008b) where movement is not only not to the edge but also in clear violation of anti-locality (cf. Grohmann 2003; Abels 2003). Since IM is not in focus at this point, I will refrain from further comments here. The only thing to be added about DiSciullo & Isac's (2008a,b) approach is that they argue that adjuncts are not subject to Merge and therefore exempt from the Proper Inclusion Condition. An assumption we will come back to presently.

SO that results from Merge an atomic object available for further Merge operations[21] in terms of a feature based account as described above. So, when abstracting away from the problems related to casting labeling in terms of a 'is-a' relation or θ-marking (cf. ibid: 58), Hornstein (2009) and Hornstein & Nunes (2008) make an interesting proposal on labeling in the context of adjunction structure.

Following proposals originally made in Chametzky (2000) and Uriagereka (2008), Hornstein & Nunes (2008: 57) capitalize on the assumption that Merge is an operation that consists of concatenation and labeling and they argue that in argument structures[22] both the concatenative and the labeling procedures apply, whereas in adjunction structures concatenation applies but labeling is not mandatory (cf. ibid: 66–67). In fact, adjunction structures then are ambiguous between those which are labeled and those which are merely concatenated without labeling. This, as Hornstein (2009: 96) shows, is what allows for making the distinction between adjuncts that are pied-piped – namely those that are labeled – and those that are stranded – namely those that are merely concatenated without labeling. Eventually, Hornstein (2009: 101–104) and Hornstein & Nunes (2008: 77–83) argue that the distinction between labeled adjuncts, which are available for further syntactic operations and merely concatenated adjuncts, which are inactive for further computations, allows for explaining why adjuncts differ from complements in some respects but pattern with specifiers in others (cf. also May 1985; Hale & Keyser 1993; 2002; Kayne 1994). In effect, the distinction between labeled and unlabeled adjuncts thus yields radically different interpretations at the C-I interface (cf. Hornstein & Nunes 2008: 83–84).

However, as Gallego (2010: 20) remarks, while it might be a reasonable assumption that adjuncts can be label-less, this does not say anything about how they are interpreted at the interfaces.[23] Also, it is not clear, what triggers the Merger of adjuncts in the first place. If label-determination as described in (5) is part

21. It should be pointed out that this is really the main motivation for Hornstein (2009: 123–125) to make use of the 'is-a' relation. Guaranteeing that concatenated SOs resulting from Merge are labeled is what ensures that the SOs are available for further concatenative processes induced by Merge. However, it should at least be mentioned that this has important repercussions for Move and for the question whether Merge takes precedence over Move (cf. ibid: 124; Chomsky 2008: 140). Hornstein (2009: 124) argues against a conception of Move as simply Copy and Remerge.

22. I use the term argument structures here as an opposite to adjunction structures which includes complement structures and specifier structures and remain noncommittal to the syntactic status of argument structure at this point.

23. This is what leads Gallego (ibid: 20–21) to argue for an analysis of adjuncts as PPs establishing a Figure-Ground relation (cf. also Talmy 2000 and Mateu 2002). Since the exact technicalities of how label-less adjuncts are interpreted at LF is not eminent for the point to be made here, I will refrain from further comment and simply assume that LF-interpretability is guaranteed.

and parcel of Merge, how then can Merge ever be triggered without labeling? The simplest answer conceivable is that all that is needed for Merge is EFs and thus merging adjuncts resonates the concatenative aspect of Merge, while labeling represents an additional step that applies only in argument structures.

Conceiving adjunct structures as unlabeled concatenates has two remarkable effects. For one, it unburdens FL from the need for relational information formerly encoded in X-bar theoretic terms that allowed for a distinction between adjunction and complementation structures (cf. Hornstein 2009: 82). Additionally, it brings to light what has been ubiquitously remarked in the literature in recent years; namely that adjunct structures are much more simple than complementation structures, which quite naturally has been reflected in their semantic properties in the long-standing tradition of Neo-Davidsonian approaches[24] (cf. among others Higginbotham 1986; Parsons 1990; Schein 1993; Larson & Segal 1995; Pietroski 2005; Hornstein & Nunes 2008; Boeckx 2008a; Pietroski & Hornstein 2009). In fact, if on the right track, this ultimately allows for completely discarding the distinction between Pair-Merge and Set-Merge.

To see how this is possible some more detailed reflections on the Set-Merge vs. Pair-Merge distinction are indispensable here. In a Bare Phrase Structure (BPS) analysis (Chomsky 1995b) traditionally Set-Merge, as described in (3), is the operation used in complementation structures (cf. Chomsky 2000: 113; 2004: 117). Pair-Merge in contrast is what is operative in adjunction structures (cf. ibid) and it crucially differs from Set-Merge in so far as the result of Pair-Merge is an ordered pair. Hence, Set-Merger of α and β leads to (7a) and Pair-Merger of α and β gives us (7b):

(7) a. $\{\alpha, \beta\}$
b. $\langle\alpha, \beta\rangle$

So in (7a) in a second step – after the concatenative procedure of Merge – the label then is determined, presumably by a labeling algorithm roughly of the sort discussed above. In (7b) on the other hand, no extra step is necessary, because the result of Pair-Merger already is an ordered pair (cf. Chomsky 2004: 117–118). This ordered pair, however, is not properly formatted for the interfaces. As Boeckx (2008a: 99) points out, in order to guarantee interface legibility, the ordered pair in (7b) needs to be turned into a monadic predicate that is readable at LF. This is done by applying a Spell-Out operation, SIMPL, to the ordered pair that turns it into an

24. It is one of the hallmarks of Neo-Davidsonian event semantics that it expresses the distinction between adjuncts, which modify an event directly, and arguments, which are associated with an event by secondary predicates (cf. Dowty 1989: 83. Hence, the sentence in (i) has the representation in (ii):

(i) Jack bought an ice-cream in West Dennis:
(ii) ∃e [buy (e) & AGENT (Jack, e) & THEME (an ice-cream, e) & in-West-Dennis (e)].

ordered set of the format {α,{α, β}}. Thus, ultimately, the product of Pair-Merge is sent to the interfaces in the same format as the product of Set-Merge. The only difference between the two operations is that Set-Merge is followed by labeling whereas Pair-Merge is followed by SIMPL.

It is immediately obvious that this conception of Merge is untenable. If FL is an optimal solution to interface requirements and if operations carried out in FL have an interface effect, then it is virtually impossible that FL entertains two operations for two conceptually very distinct phenomena and hands them to the interfaces in the same format.

Boeckx (2008a: 100–101) provides an alternative and ultimately argues for a reversal of the two operations so that Set-Merge is what yields adjunction and Pair-Merge is what results in complementation. Boeckx's reasoning is twofold. First, he points out that adjunction structures are much more flexible than complementation structures and ties this to the lack of labeling argued for in Hornstein & Nunes (2008) and Hornstein (2009). Second, this absence of labeling coincides with the absence of a Probe-Goal relation that is fundamental for any kind of featural transaction:

> Adjuncts merge with their hosts under *Match* (like any instance of *Merge*), but the matching relation here is distinct from the matching giving rise to valuation [...] The absence of checking would account for the optional character of adjunction. (Boeckx 2008a: 101)

Merge so defined may be more fit to account for the complement vs. adjunct distinction, but it raises some questions. If Set-Merge, as the operation for adjunction, is an instance of feature-Matching without Valuation, in how far is this different from Merger of T or v_{def} (cf. Gallego 2010: 27 and footnote 15)? And how is the product of Set-Merge different from the product of Pair-Merge at the interfaces? Or put differently, why is it that Set-Merge, as the operation for adjunction that is not labeled, can be processed at the interfaces at all?

Boeckx (2008a) does not explicitly consider the first question but provides an answer to the second one that explains the difference between the Merger of defective core functional categories (cf. Chomsky 2000; 2001) and the Merger of adjuncts. Under his analysis adjunct structures *per se* pose a 'mapping problem' to the interfaces that can be dissolved only when adjunction is understood in terms of Later-Merge (cf. Boeckx 2008a: 102–103). This is reminiscent of ideas already floated as early as Lebeaux (1988) and Chomsky (1995a). Hence:

> [U]nlike complementation, where both Merge members are introduced at the same time (symmetry/order independence of Merge), adjunction attaches an element to an already present element. [...] T[his] temporal asymmetry [...] is what allows Set-Merge representation to be mapped onto SEM/PHON. (Boeckx 2008a: 102–103)

So adjunction boils down to instances of Set-Merge where an already existing SO is concatenated with another LI or SO at a later stage of the derivation. Following Boeckx (ibid) adjunction then can be represented as follows:

(8) $\{\{\alpha\}_t, \{\beta\}_{t+1}\}$

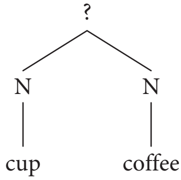
This conception of Set-Merge for adjunction structures paired with the conception of Pair-Merge for complementation structures that has been discussed above should lead to unambiguous Merge that returns an asymmetric output in all cases. However, the idea to define Set-Merge as an instance of Later-Merge as in Lebeaux (1988) has certain consequences. It suggests that adjunction always applies at a late(r) stage of the derivation. All instances of First-Merge then must be instances of complementation.

So far so good, as long as there are no counterexamples to First-Merge being Pair-Merge, this is an assumption that can be maintained. What is not clear though is when exactly Later-Merge applies. Does Later-Merge mean that the derivation must have built a SO that is sent to the interfaces before an adjunct that does not produce a new label can be added? If so, how are these adjuncts integrated? Are they constructed in FL and kept in a separate plane (cf. Uriagereka 1998; 2008; Chomsky 2004) so that they can be attached to the SO created by FL in a post-syntactic but pre-Spell-Out component? If so, where is this component located and how is it accessed and regulated? If not, how do the interfaces keep track of the temporal component of Later-Merge? And how can the interfaces process a structure that is label-less?

1.3 Outlook

Finding an answer to these questions that is in line with the SMT and basic assumptions of minimalist theorizing is the aim of this study. Three types of structures in which Merge produces a SO that is not labeled are looked at: nominal root compounds, incorporated nominal gerunds (which are compared against their non-incorporated counterparts) and small clauses. In the derivation of each of these structures a point of symmetry (PoS) arises that is not broken spontaneously (cf. Boeckx 2008a: 88) in the course of the derivation. In the nominal root compounds this PoS arises after the first application of Merge, which combines two nominal heads of which neither one immediately projects:

(9)

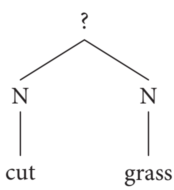


(coffee-cup)

Nominal root compounds are typically analyzed as adjunction structures (cf. e.g. Roeper, Snyder & Hiramatsu 2002; Roeper & Snyder 2005) and in so far seem to fit with the description of label-less concatenates provided here. The eminent question however is, how the (First-!)Merger of two nominal heads can lead to a labeled SO that is available for further computation. Chapter 2 discusses several approaches to the derivation of these structures and argues that none provide an answer compatible with the conceptions of Merge and the requirements imposed by the SMT. An alternative analysis that is not only in line with the SMT, but also allows for a principled distinction between productive compositional recursive compounds and their unproductive, non-compositional and non-recursive counterparts, is presented. The distinction between compositional compounds and their non-compositional counterparts, it turns out, is not based on inter-language variation of the type advocated for e.g. in Delfitto, Fábregas & Meloni (2008) but rather on inter-language variation that is fine-tuned by cyclic derivation and does not need to make use of additional stipulative assumptions or unmotivated features.

Another instance where First-Merge creates a point of symmetry that needs to be dissolved is in incorporated nominal gerunds. Just as in (9) Merge here combines two heads, however, these heads rather than being adjuncts presumably stand in a complement relation where the verbal head selects the nominal head:

(10)



(grass-cutting/s)

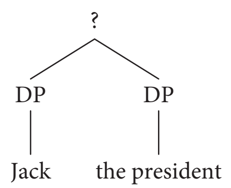
What is not clear though is how this can be accounted for under the φ-feature driven account to Merge and labeling discussed above, nor why incorporation is forced in these structures and why simple feature-Matching and Valuation is not enough. In particular the lack of an effect on interpretation between the incorporated (11b) and the non-incorporated structure (11a) and the clear effect on interpretation between pluralized (11c/d) and non-pluralized versions of nominal gerunds (11a/b) is puzzling in this context:

(11) a. the cutting of the grass
b. the grass-cutting
c. the grass-cuttings
d. the cuttings of the grass

In Chapter 3 these structures are analyzed in minute detail and it is shown that current approaches to gerunds and their incorporated versions (cf. e.g. Barrie 2006; Iordăchioaia & Soare 2008; Alexidaou; Iordăchioaia & Soare 2010) can neither accurately capture the observed phenomena above, nor do they comply with the account on Merge developed here. Again, an alternative analysis that does not rely on additional mechanisms but strictly adheres to the SMT, which guides a cyclic approach to derivation that is motivated by independent principles, is presented.

Chapter 4 looks at one more instance of Merge that produces a symmetric outcome: small clauses (SCs). Following the analysis in Moro (2000; 2007) the asymmetry here does not result from the Merger of two heads but two phrases instead:

(12)



(they made sc [Jack the president])

In principle, this structure is reminiscent of the adjunction structures resulting from the Merger of two phrases as discussed above. However, as den Dikken (2006) argues, SCs are predication structures rather than modification structures. This, of course, begs the question how this is accommodated under the SMT based approach described here. The analysis provided in Chapter 4, which is in part based on the approach outlined in Bauke & Roeper (2012), does not only provide an answer to the question of how the label of the SC can be determined but it also shows in which way a Phase-based approach to the derivation of SCs allows for precise predictions on relevant Phase-boundaries that indicate designated positions that the SC constituents can be moved to.

So, the investigation of the structures in Chapters 2–4 illustrates that the adjunction structures in (9), the complementation structures in (10)–(11) and the predication structures in (12) can be uniformly accounted for under a Phase-based approach to the SMT that does not rely on the stipulation of additional *ad hoc* features or mechanisms. In fact, only independently motivated assumptions that follow naturally from interface requirements can not only provide the basis for an analysis of the structures discussed here, but they also have interesting implications for the assessment and development of minimalist theory. These implications are discussed in Chapter 5 and include the following points.

First, the question whether Phase-based computations impose restrictions on lexicon-syntax interaction (cf. e.g. Boeckx 2008a) or on syntax-interface interactions emerges (cf. also Gallego 2008: 41). The analysis in this study certainly points into the latter direction, but without negating the relevance of the former and the answer to the question includes the discussion of what constitutes a Phase. While it is pretty clear what constitutes a Phase in the clausal domain (cf. e.g. Chomsky 2000; 2001; 2004; 2005; 2008), i.e. φ-feature checking, as well as how this follows from SMT considerations and what counts as a relevant Phase-head in the clausal domain, i.e. **v* and C, it is far less obvious what closes off a Phase (cf. e.g. Chomsky 2000 vs 2001; Richards 2007; Grewendorf & Kremers 2009; Gallego 2010). Additonally, neither the status of Phases at the subphrasal level (i.e. word-level) (cf. Marantz 2001; 2007) and in the nominal domain (cf. among many others Svenonius 2004; Bošković 2005; Hiraiwa 2005; Matushansky 2005; Müller & Heck 2008; Ott 2008; Samuels 2008; Richards 2012) is unanimously agreed upon in the literature, nor is clear what the status of Non-Phase-heads is (cf. in particular Chomsky 2008 vs. previous accounts) and whether Phases can be slided or extended (den Dikken 2007a;b; Gallego & Uriagereka 2007; Gallego 2010).

On the basis of the analysis in Chapters 2–4, it is argued that Phases play a crucial role at word-level syntax and in the nominal domain and indeed are guided by the same principles as in the phrasal domain. With respect to the question of Phase-sliding/Phase-extension, which necessarily requires head-movement, it is also discussed, whether head-movement is a purely phonological operation as Chomsky (2000) argues or whether alternative accounts of head-movement either as remnant movement (cf. e.g. Koopmann & Szabolcsi 2000; Nilsen 2003) or (even) as a syntactic operation (Harley 2004; 2009; Donati 2006; Matushanksy 2006; Cechetto & Donati 2010; Gallego 2010; Roberts 2010) are possible and how the problems of tucking-in and conflation fare under these accounts. Again, the analysis of the structures in the preceding chapters guides the discussion here.