Introduction

This research survey combines an introduction to Phase Theory with an assessment of the state of the art in Phase Theory. The term Phase Theory refers to a set of theoretical innovations in post-2000 minimalism (Chomsky 2000, 2001, 2004, 2005, 2008).[1] One of the core ideas in minimalism is the idea that the language faculty is an optimal solution to the constraints imposed on it by the two cognitive systems with which it interacts: the system of thought and the articulatory–perceptual system. What Phase Theory adds to this picture is the idea that the language faculty interacts with these two cognitive systems at very specific points during the syntactic derivation, and, consequently, that syntactic derivations are constructed in chunks referred to as **phases**.[2] In most general terms, phases cannot be accessed by the narrow syntax once they are transferred to the interfaces.

My goal in this survey is to combine an introduction to a given issue within Phase Theory with an overview of the existing research on this issue (and an assessment thereof), giving the reader a sense of what is fairly settled upon and what is still under debate.[3] The fact that there is a lot of research that relies on phases shows a need for a survey that situates phases in current syntactic theory, introduces the technical details of Phase Theory, synthesizes the existing research

[1] Chomsky in his writings is very careful about distinguishing a program from a theory, emphasizing the programmatic nature of minimalism (see in particular Boeckx 2006 for a more detailed discussion of this distinction). Chomsky does not use the term Phase Theory in his early writings on phases but does so more recently: 'One goal of *Phase Theory* is to provide the mechanisms to distinguish copies from repetitions, as straightforwardly as possible' (Chomsky 2012: 3).

[2] The resulting model is also sometimes referred to as a Multiple Spell-Out (MSO) model. The idea of Multiple Spell-Out goes back to Uriagereka (1999) (see also Uriagereka 2012 for a fuller, book-length exposition).

[3] There are a number of monographs, volumes and collections that focus on various aspects of Phase Theory, which I build on and am intellectually indebted to (see, among others, Frascarelli 2006, Gallego 2010, 2012, Grohmann 2009a, 2009b, 2009c, McGinnis & Richards 2005, and the individual contributions in the two issues of *Linguistic Analysis* 33, guest-edited by Kleanthes Grohmann).

on phases (pointing out issues that might be still contentious), outlines directions for future research, and, last but not least, standardizes the notation.

Even though many (though not all) syntacticians (explicitly or implicitly) assume the concept of a phase, there seems to be less of a consensus regarding many of the most fundamental properties of phases, such as those listed in (1).

(1) a. How do we define phases?
b. What categories count as phases?
c. Do the same categories count as phases with respect to semantic and phonological considerations?
d. Are phases dynamic or static?
e. Is there any crosslinguistic variation with respect to phasehood?
f. How do phases interact with the interfaces?

The fundamental question to answer before we can even begin to address some of the questions listed above is in what sense a syntactic theory that involves phases is more adequate (in a descriptive, explanatory, or beyond-explanatory adequacy kind of sense) than a syntactic theory that does not involve phases.[4] This is the question we will be coming back to throughout the book. In the remainder of this introduction, I provide a brief summary of each chapter.

Chapter 1 'The Minimalist Program' provides an overview of the core aspects of the Minimalist Program. It outlines the general architecture of the minimalist grammar, and lays the groundwork for the discussion in the following chapters by focusing on the concepts that will be crucial to the understanding of phases, such as the distinction between interpretable and uninterpretable features or the concept of Spell-Out. This chapter is not meant as an introduction to (or survey of) minimalist syntax; nevertheless, readers less familiar with minimalism will find all the necessary concepts, terms and mechanisms introduced in this chapter.

Chapter 2 'Motivating phases' turns to phases themselves. It introduces the concept of a phase, situating phases in the context of current minimalist approaches to syntactic dependencies, and asking if syntactic theory with phases is more adequate than a theory without phases, or a theory in which all phrases are phases. This chapter also gives a historical perspective on phases, and addresses some of the criticisms that have been levied in the literature against them, such as Boeckx & Grohmann's (2007) critique of phases as 'barriers in disguise'.

Even though the idea of a cycle, which is conceptually related to a phase, goes back to the early days of generative grammar, the current concept of a phase first appeared in Chomsky's (2000) 'Minimalist Inquiries', where phases (to be more specific, lexical subarrays associated with phases) were introduced as a solution to a problem arising from the so-called Merge over Move principle. Since then,

[4] Following Chomsky (2004), I take the term *beyond explanatory adequacy* to refer to the *why*-questions about language, captured by the following quote: 'In principle, then, we can seek a level of explanation deeper than explanatory adequacy, asking not only *what* the properties of language are but also *why* they are that way' (Chomsky 2004: 105)

much research has focused on defining phases and formulating independent phasehood diagnostics. The existing definitions of phases I survey in this chapter are listed in (2).[5]

(2) a. Phases are propositional objects.
b. Phases are convergent objects.
c. Phases are objects interacting with the interfaces.
d. Phase heads are loci of uninterpretable features.
e. Phases are predication structures.
f. Phases are phrases.

From a diagnostic perspective, perhaps the most important aspect of Phase Theory is the so-called Phase Impenetrability Condition, which deems a portion of a phase impenetrable or inaccessible to operations from the outside. This chapter also surveys the various versions of the Phase Impenetrability Condition proposed in the literature, focusing on the empirical predictions they make, and ways to unify them (see Müller 2004, Richards 2004, 2011, among others). The Phase Impenetrability Condition is tightly linked to the concept of Multiple Spell-Out, which I also elaborate on in this chapter, sorting through the logical possibilities of how Multiple Spell-Out can proceed, i.e. spelling out to the two interfaces at different points in the derivation, for example. Finally, this chapter introduces the concept of Feature Inheritance, as developed by Chomsky (2008) and Richards (2008), which is a logical consequence of defining phase heads as hosts of uninterpretable features. If uninterpretable features are a property of phase heads, the only way non-phase heads can get them is via Feature Inheritance.

Chapter 3 'Phasehood diagnostics' turns to the many diagnostics that have been proposed in the literature, a subset of which will serve as the basis for the discussion of specific phases (CPs, *v*Ps, DPs, PPs etc.) in the chapters that follow, and the arguments in favor of (or against) these categories being phases. A common thread in many existing characterizations of phases is that they should exhibit a certain amount of independence and coherence at the interfaces. This, however, only raises the question of what it means for a given category (a candidate for a phase) to be semantically or phonologically independent and coherent. Furthermore, are there any phasehood diagnostics that do not fall neatly into either of the two groups (PF versus LF diagnostics): purely syntactic or purely morphological diagnostics? Given such rather vague existing characterizations of phases, this chapter focuses on the more tangible questions that can be (and have been) asked to establish the phasehood of a given category, which I list in (3) below. It examines these questions with a critical eye towards establishing genuine phasehood diagnostics, and avoiding those that might instead be diagnosing something other than phasehood (such as constituency or phrasal status).

[5] See also Boeckx (2006), Boeckx & Grohmann (2007), Den Dikken (2007) and Gallego (2010) for a discussion of these different views of phases, and of the problems some of them raise.

(3) a. Is XP a domain for feature valuation?
b. Is X the locus of uninterpretable features?
c. Does X trigger Transfer?
d. Is XP a phonological domain?
e. Can the complement of X be elided?
f. Can XP be moved?

The Phase Impenetrability Condition also gives rise to a number of tangible phasehood diagnostics, coming mostly from the realm of successive cyclic movement through the edge of the phase, which in turn can be diagnosed by affirmative answers to the following questions:

(4) a. Can the moved element be interpreted at the edge of the phase?
b. Can the moved element be pronounced at the edge of the phase?
c. Can the moved element leave something behind at the edge of the phase?

The discussion of phasehood diagnostics also raises the question of whether there is any crosslinguistic variation with respect to phasehood. This is the issue which I come back to in Chapter 6. While variation with respect to whether a given language has phases or not seems highly unlikely and implausible, given the conceptual arguments in favor of having phases to begin with (such as reducing computational load and being independently motivated by the interfaces), it is certainly possible for languages to differ with respect to what categories count as phases.

Chapter 4 'Classic phases' discusses in detail three categories that are commonly assumed to be phases – CPs, *v*Ps and DPs – and applies the diagnostics established in Chapter 3 to these categories.[6] The phasehood status of CP is relatively easy to establish: the evidence in favor of successive cyclicity from the literature on A-bar dependencies is typically taken as evidence for CPs being phases (see, for example, Lahne 2008 for an illuminating overview). The evidence includes phenomena like wh-copying (Felser 2004, Manetta 2010, McDaniel 1989, among many others), scope marking (Dayal 1996, Lutz, Müller & Von Stechow 2000, Stepanov 2000, among others), complementizer agreement (Carstens 2003, Carstens & Diercks 2011, Haegeman 1992, Haegeman & Van Koppen 2011, Zwart 1993, 1997), wh-quantifier stranding (McCloskey 2000, 2001), reconstruction (Barss 1986, 2001) and left branch extraction (Wiland 2010). The evidence in favor of *v*Ps being phases is similar in spirit. This chapter also reviews the debate on whether unaccusative and passive *v*Ps constitute phases, as argued for by Legate (2003), and against by Den Dikken (2006a). While many of the facts that are typically deemed to bear on the issue of C or *v* being a phase head might be given

[6] The discussion in Chapters 4 and 5 is a sequence of case studies. There are other categories that are conspicuously absent from the discussion here (APs, AdvPs, various functional projections in the left periphery of a clause) whose phasehood we might wonder about. I thank Kleanthes Grohmann for bringing them to my attention.

alternative explanations that do not necessarily rely on movement through the specifier of CP or *v*P, such accounts typically still posit a relationship between C (endowed with uninterpretable features of the requisite sort) and the wh-pronoun in its domain. This also points towards C being a phase head, given that only phase heads are assumed to be the loci of uninterpretable features. More generally, I hope to show in this chapter that phase-theoretical accounts have the advantage of establishing connections between sets of facts that otherwise remain isolated and require independent explanations. For example, why should complementizer agreement phenomena and locality restrictions on movement involve C? Or why would Austronesian extraction restriction and constraints on parasitic gap formation be sensitive to the properties of little *v*? Granting these projections a privileged syntactic status (namely, the status of a phase) brings us closer towards understanding why syntactic phenomena should cluster around them.

The idea that DP might be a phase as well, explored by Matushansky (2005), Hiraiwa (2005) and Svenonius (2004), among others, should not come as a surprise, given the many structural and interpretive parallels between CPs and DPs, discussed in the literature going back to the very early days of generative grammar. However, since CPs contain other phases (namely *v*Ps), an interesting question is whether DPs contain other phases as well. In order to tackle this question, this chapter also addresses the internal structure of DPs, motivating the need for DP-internal projections such as NumberP, PersonP or ClassifierP, and asking which of them, if any, might be phases as well.

Chapter 5 'Other ph(r)ases' turns to categories whose phasehood status is somewhat more controversial, and still debated in the literature: Predication Phrases, Prepositional Phrases and Applicative Phrases. All of them have been argued to constitute phases (see, for example, Abels 2003, Radkevich 2010 on PPs as phases, and McGinnis 2001 on Applicative Phrases as phases); yet they are not considered to fall into the widely accepted phasehood canon. What makes these categories somewhat more controversial is that many other questions about them have to be answered first before their phasehood can be entertained. For example, if phasehood is a property of functional categories, the question that needs to be resolved for prepositions is whether they are functional or lexical (or both or neither, depending on the preposition).

Chapter 6 'Variation in phasehood' takes up the issue of whether there is any crosslinguistic variation with respect to phasehood. There are two questions to consider here: the question of whether non-phase heads can acquire phasehood status in the course of the derivation (and conversely, whether phase heads can lose their phasehood status in the course of the derivation), and the question of whether different categories can count as phases with respect to phonological and semantic considerations. The former scenario (a head becoming a phase or ceasing to be a phase) has been argued to arise as a result of head movement (Phase Extension of Den Dikken 2007 or Phase Sliding of Gallego 2010). The latter scenario (a category being a PF phase but not an LF phase or vice versa) has been explored by Marušič (2005) as a way to handle total reconstruction and

covert movement and by Felser (2004) to handle wh-copying. This chapter also addresses crosslinguistic variation with respect to phasehood: if phases are dynamic and head movement can extend phasehood, a certain amount of variation will come from independent considerations (such as the presence or absence of certain types of head movement). Variation in phasehood can also follow from variation in lexical inventories.

Chapter 7 'Phases at the interfaces' examines the roles phases play at the interfaces, putting them in the more general context of the syntax–phonology and syntax–semantics interface. With respect to the PF interface, it focuses on the questions of whether phases (or Spell-Out domains) are relevant and substantive phonological units, and how these phasal or Spell-Out units are manipulated by phonology. This chapter examines the role phases play in determining linear order (as in Fox & Pesetsky's (2005) Cyclic Linearization) and nuclear stress (see Adger 2007, Kahnemuyipour 2003, 2004, 2005, Kratzer & Selkirk 2007, among others).

The potential evidence for the significance of phases at the syntax–semantics interface comes not only from phenomena like scope ambiguities (on the assumption that Quantifier Raising is constrained by phasehood) and the propositional status of phases – both of which feature prominently as phasehood diagnostics – but also from the idea that the boundary between *v*P and CP phasal domains corresponds to the distinction between nuclear scope and restrictive domain in the tripartite quantificational structure, as proposed explicitly by Biskup (2009a).

Chapter 8 'Summary' provides a brief summary and conclusion.

1

The Minimalist Program

1.1 General architecture

The current chapter offers a bird's eye view of the Minimalist Program. It is not meant as a comprehensive introduction (or a thorough overview) of minimalism. Rather, its goal is to give readers less familiar with minimalism the necessary and sufficient background to follow the discussion of phases in the rest of this book. For the sake of clarity, the technical terms that I will be referring to throughout the book will be given in bold when they are first introduced. For a more thorough introduction to minimalism, I refer the interested reader to Adger (2003), Hornstein, Nunes and Grohmann (2005) and Chapter 1 of Gallego (2010).

What came to be known as the Minimalist Program was articulated explicitly in the early nineties with the publication of works such as Chomsky's (1991) 'Some Notes on Economy of Derivation and Representation', and his (1993) 'A Minimalist Program for Linguistic Theory', both of which later became two of the four chapters of Chomsky's (1995) *The Minimalist Program*. As Chomsky emphasizes in his writings, minimalism is grounded in the Principles and Parameters model, which gave us the beginnings of an understanding of which properties of language are universal (and perhaps unique to it), and which ones are subject to crosslinguistic variation. This, in turn, led to deeper questions, which are at the core of minimalist theorizing nowadays. These are questions that go beyond explanatory adequacy, alluded to above, such as the question of why language is the way it is. Computational efficiency and interface conditions play a central role, as stated succinctly in the following quote from 'Beyond Explanatory Adequacy'.

> (1) Its [the Minimalist Program's, B.C.] task is to examine every device (principle, idea, etc.) that is employed in characterizing languages to determine to what extent it can be eliminated in favor of a principled account in terms of *general conditions of computational efficiency* and *the interface condition* [emphasis mine, B.C.] that the organ must satisfy for it to function at all.
>
> (Chomsky 2004: 106)

This is also clear in Chomsky's discussion of the so-called three factors in language design and three types of conditions in language acquisition (see Chomsky 2005 in particular). These are listed in (2a–c), and they help determine how a child gets from the initial state (S_0) of linguistic competence to the final state: the fully formed adult state of linguistic competence. (2b) are the interface conditions, and (2c) are the general properties of efficient computation.

(2) a. unexplained elements of S_0
b. IC (the principled part of S_0)
c. general properties (Chomsky 2004: 106)

The minimum that the language faculty (**FL**) has to accomplish is to interface with language-external systems. The two external systems in question are the **Sensorimotor** (**SM**) system and the **Conceptual-Intentional** (**C-I**) system. The conditions imposed by these two external systems are referred to as **Legibility Conditions**, **Bare Output Conditions** or **Interface Conditions (IC)**.[1] The **Strong Minimalist Thesis** (**SMT**), given in (3), states that language is designed to interface with the external systems in an optimum way.[2]

(3) The substantive thesis is that language design may really be optimal in some respects, approaching a "perfect solution" to minimal design specifications.
(Chomsky 2000: 93)

The general architecture of the language faculty is as follows. Language has three components: **Narrow Syntax** (**NS**), the phonological component Φ and the semantic component Σ.[3] For the most part, we will be concerned here with Narrow Syntax and its computational processes.

Each derivation starts with a set of 'lexical items' which are manipulated in the course of the derivation by the syntactic operations **Merge** and **Agree**. I will discuss these two operations in more detail in Sections 1.2 and 1.4, respectively. The 'lexical item' is, strictly speaking, a bundle of features, not a primitive syntactic object (hence the quotes).[4] This set of lexical items is called a **Lexical Array** (**LA**) and is represented as an unordered set. A Lexical Array augmented by information

[1] To the best of my knowledge, these terms are used interchangeably.

[2] The formulation of the Strong Minimalist Thesis in 'Minimalist Inquiries' (which is the one given in (3)) is slightly different from the one Chomsky gives in 'Beyond Explanatory Adequacy', where he formulates it as in (i):

(i) The set of unexplained elements of S_0 is empty. (Chomsky 2004: 106)

S_0 refers to the genetically determined initial state in the process of language acquisition, which is what UG provides.

[3] There is no PF or LF cycle and thus there are no PF or LF operations. The terms are used to refer to PF or LF representations. The terms PF interface and LF interface are used to refer to the interface with SM or C-I systems, respectively.

[4] This assumes something like the Late Vocabulary Insertion model of Distributed Morphology of Halle & Marantz (1993) and much later work.

on how many times each lexical item is selected from the lexicon is called a **Numeration** (**N**). This information is represented by subscripts. In simple cases, Numerations and Lexical Arrays are equivalent, as shown in (4b–c).

(4) a. Icarus likes nuts.
b. LA = {Icarus, likes, nuts, *v*, T, C}
c. N = {Icarus_1, likes_1, nuts_1, v_1, T_1, C_1}

The two diverge when a single item is used more than once in a given derivation, as in the infamous example given in (5a).[5]

(5) a. Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo.
(Pinker 1994: 2006, crediting Annie Senghas)
b. LA = {Buffalo, buffalo, buffalo, C, *v*, T}
c. N = {Buffalo_3, buffalo_2, buffalo_2, C_2, v_2, T_2}

However, one does not need to resort to exotic examples of this sort to show how Lexical Arrays and Numerations differ; relatively simple sentences with any level of embedding make the same point:

(6) a. Icarus thinks he likes nuts.
b. LA = {Icarus, thinks, he, likes, nuts, *v*, T, C}
c. N = {Icarus_1, thinks_1, he_1, likes_1, nuts_1, v_2, T_2, C_2}

The output of a Narrow Syntax derivation is a pair of representations <PHON, SEM>, which is accessed by the two interfaces defined above: the SM and the C-I interface. The derivation **converges** if the two representations satisfy the conditions imposed by the two interfaces; otherwise it **crashes**.[6] For a given representation to meet the Interface Conditions simply means it has to be legible to the external systems; hence the term Legibility Conditions. The question of what it means to be legible at a given interface is not trivial. A common and intuitively correct understanding of this concept of legibility is the following: an expression is legible at a given interface level (PF or LF) only if it consists of features that can be interpreted by the language-external systems: the SM and C-I system, respectively. But, of course, making convergence contingent on the presence of features that can only be interpreted at the interfaces raises an obvious question of what features the two interfaces can interpret. It seems quite plausible to assume that

[5] Its notoriety comes from lexical ambiguity, of which it provides a very extreme illustration, not from the distinction between a lexical array, which in this case contains three distinct lexical items 'buffalo' (i.e. the proper name *Buffalo*, the common noun *buffalo*, and the less commonly used transitive verb *to buffalo* meaning *to bully*) and the Numeration that includes multiple occurrences of each of them. The following paraphrase helps distinguish the different meanings of *buffalo*:

(i) The buffalo from Buffalo that (another) buffalo from Buffalo bullies himself bullies (yet another) buffalo from Buffalo.

[6] See, however, Frampton & Gutmann (2002) for a proposal that syntax only generates convergent derivations, and Preminger (2011) for a proposal that having unvalued features does not always lead to a crash.

the SM interface can interpret features having to do with linear order (if such exist), syllable structure, prosodic structure or intonation. The C-I interface, on the other hand, should be able to interpret features having to do with scope, quantification, referentiality, specificity, propositional status etc. Neither interface can interpret formal features, such as structural case features or categorial features. Since features play such a major role in minimalism and there are quite a few contentious issues surrounding them, we will devote an entire section to them (Section 1.3).

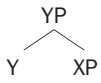
The three basic operations that manipulate lexical items selected from the lexicon are **External Merge**, **Internal Merge** and **Agree**. These three, in conjunction with a more detailed discussion of features, are the focus of the next three sections.

1.2 External and Internal Merge

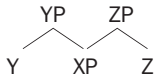
Recursion, the property of language that allows smaller units to combine iteratively to form larger units forming hierarchically structured objects, and displacement, the property that gives us the intuition that syntactic objects can surface in one position but be understood as belonging in another position, are two very fundamental (perhaps the most fundamental) properties of language. Chomsky (2004) distinguishes two kinds of Merge, **External Merge** (**EM**) and **Internal Merge** (**IM**), to capture these two fundamental properties. External Merge is the basic concatenation operation responsible for recursion in language. It takes two objects (such as X and Y in (7a), which have been first selected from the Numeration), and combines them into one bigger object, as shown in (7b).[7] External Merge is a recursive operation; one of these two objects could

[7] Elsewhere, I have argued that Merge can also create structures in which a single object can end up shared between two objects, referring to this type of Merge as Parallel Merge (see Citko 2011b and the references therein), illustrated in (i–ii). Parallel Merge combines the properties of External Merge and Internal Merge. Before Merge takes place, Z and YP are disjoint (as in External Merge) but Z merges with a subpart of YP (as in Internal Merge).

(i) Merge X and Y, Project Y



(ii) (Parallel) Merge XP and Z, Project Z



Chomsky (2007) excludes such a possibility on the grounds that 'it requires new operations and conditions on what counts as a copy, hence additional properties of UG' (Chomsky 2007: 8, note 10).