

CHAPTER 6

PICKING UP THE PIECES: A RADICAL VIEW FROM THE LEFT

This final chapter—partly as a conclusion of this work, partly as a road map for further research—summarizes the main properties of the framework of Prolific Domains, clausal tripartition and Exclusivity (the theory of anti-locality) which I have proposed and presented in the preceding chapters. In so doing, I elaborate on two concepts introduced earlier, address and context, and lay out a possible route for the interaction between syntactic computation and interaction with the interfaces. In this context, similarities with and differences to recent approaches will be discussed.

6.1 Prolific Domains: The Framework

In this dissertation I have proposed a theory of anti-locality effects, the fact that while subject to locality conditions, movement must not proceed too locally. The clue to define anti-locality lies in an appropriate domain of evaluation relevant for closeness. Minimalist inquiries into syntactic computation offer a natural cut: the part of the clause where theta relations are created, the part that licenses agreement properties and the part responsible for those types of discourse information that are syntactically relevant. This

tripartition is natural in the sense that it is defined over parts of a phrase marker which are characterized by checking configurations of formal features pertaining to each of these three “super-features,” and there seems to be evidence, all other things being equal, that no element checks more than one feature of these three types. We called each of these three areas a Prolific Domain, an area within the phrase marker for a particular clausal computation (“domain”) which itself consists of a number of projections (“prolific”). We labeled the first Prolific Domain θ -domain, the second ϕ -domain and the third ω -domain. This is repeated here from (26) in chapter 2:

- (1) *The concept of Prolific Domains ($\Pi\Delta$)*
- i. θ -domain: the part of the derivation where theta relations are created
 - ii. ϕ -domain: the part of the derivation where agreement properties are licensed
 - iii. ω -domain: the part of the derivation where discourse information is established

We considered to think of the three “super-features” as context values, a notion to be refined momentarily. This helped us formulate the theory of anti-locality in terms of Exclusivity, with the definition of (30) from chapter 2 repeated here for convenience:

- (2) *Condition on Domain Exclusivity (CDE)*
- An object O in a phrase marker must have an exclusive Address Identification AI per Prolific Domain $\Pi\Delta$, unless duplicity yields a drastic effect on the output.
- i. An AI of O in a given $\Pi\Delta$ is an occurrence of O in that $\Pi\Delta$ at LF.
 - ii. A drastic effect on the output is a different realization of O at PF.

Before addressing the notions “address” and “context” further, occurrences of O within a Prolific Domain was evaluated over non-distinct copies. As a rough guide, we can thus take Exclusivity to be the ban on XP-movement within a Prolific Domain. The notion of Prolific Domain, in turn, was understood in terms of the following working hypothesis, which will be fleshed out further in this chapter ((27) from chapter 2):

(3) *Prolific Domain* (working hypothesis)

A Prolific Domain $\Pi\Delta$ is a contextually defined part of the computational system

- i. which provides the interfaces with the information relevant to the context and
- ii. which consists of internal structure, interacting with derivational operations.

These are the cornerstones of the framework sketched in the course of this work. They allowed us to ban movement of a given element from one position (launching site) to another position (target site), where the target site would be too close to the launching site. All in all, this makes the right predictions: lexical items do not move from one θ -position to another within the same clause, they do not move several times within the same agreement layer to check several sets of ϕ - or Case-features, they do not move within an articulated COMP to check off a set of topic-, focus- and/or Wh-features, and so on. Movement is understood in the usual way (Chomsky 1993), in terms of Copy and Merge, where the lower copy is subsequently deleted on principled grounds (Nunes 1995, 1999). Nunes in fact shows that deletion of non-distinct copies is forced by the phonological component: in order to linearize the objects of a phrase marker at PF, where each object must be asymmetrically c-commanded by the one preceding it. If an object is part of a dependency consisting of two or more non-distinct copies, the lower copies must be deleted, or else the non-distinct copies would at the same time precede and follow each other. Assume this much to be correct, roughly as outlined here.

Non-distinctness of copies turned out to play a major role in our framework also. We found that while it holds in general, anti-locality seems to be violated in some cases—namely those that involve a dependency between a phrase and a pronoun of sorts in which both were arguably non-distinct copies of one object in the phrase marker at some point in the derivation (identity, connectivity etc.). The approach suggested here is that in these instances, the pronominal element is a spelled out copy of the phrase (“Copy Spell Out”), and that it seems to bear all characteristics of a deleted copy, except for its PF-matrix: it is c-commanded by the original copy, it is the lower of two copies, and so on.

If lower non-distinct copies need to be deleted for PF reasons in the case of Move, we might want to consider similar motivation to spell them out in other instances. The definitions of Prolific Domains and the Condition on Domain Exclusivity suggest exactly that. We have assumed this to be the case throughout, and it did some work for us. Let us now look at some relevant details a little bit closer. The next section is thus an extension of the core proposal from chapter 2. It is not the only possible conceptualization of the idea, and other ways are not excluded, but it is one which seems appropriate in the context of recent work on the syntactic computation and its connection to the interfaces. To mind comes in particular recent work by Chomsky (1998, 1999), Epstein et al. (1998), Boeckx (1999a), Uriagereka (1999b), or Platzack (in press), for example.

6.2 Syntax at the Interfaces: A Sketch

The tripartition envisioned here (and elsewhere; cf. Rizzi 1997, Platzack, in press and many others) is intuitive, and given the work it does, we can accept its existence with little further ado. What needs to be addressed are the details. Many questions have to be left open, but I will sketch a possible approach concentrating on the ingredients of the definitions repeated above and putting them into some perspective.

One question that will not be addressed is why these three and only these three “super-features” should exist and play such a crucial role for the computation.¹ For what-

¹ Another questions regards the number of the Prolific Domains. I do not exclude other possible Prolific Domains, possibly in form of small clauses, or on a different level than the clause (such as the nominal level.). It is likely that DP constitutes another Prolific Domain, as has often been observed that the nominal layer and the full clausal layer share a number of properties.

Returning to the clausal level, Rizzi (2000) also considers “super-features,” though from the other end of locality, the upper bound on dependencies in terms of Relativized Minimality (basically updating the framework of Rizzi 1990).

ever reason, they do. But we can ask ourselves how they come into play. As our working hypothesis says, a Prolific Domain is contextually defined. Let us take each of these contextual “super-features” to be a context value. We thus deal with three relevant context values, represented as $|\theta|$, $|\phi|$ and $|\omega|$. Anything confined in the domain marked by one of these context values is part of that Prolific Domain. The possibly simplest way to integrate such values into the system—without any additional devices—is by treating them as part of the lexical make-up of the relevant lexical or functional head, rather than indexicalizing relevant material with this context information, for example. One view of the initial make-up of these heads is that they contain at least the formal feature they correspond to, which can be an abstract morpheme or a phonetically realized element. The particular details depend on the theoretical assumptions regarding the initial structure, the exact content of the lexical array LA, the role of morphology, and possibly more.² Under this view, whether a particular head belongs to one or the other Prolific Domain is universally determined. Thus, V and *v* are always part of the θ -domain because both contain $|\theta|$, Agr or T always belong to the ϕ -domain and FocP or CP to the ω -domain. This is desirable insofar as the observations on anti-locality laid out in chapter 2 arguably hold across languages (that is, no element in a simple, monoclausal construction checks two θ -features, several Cases, and so on).

We now have accounted for the base of the definition of a Prolific Domain (3) and the specification (3ii), given more or less standard assumptions about the lexicon LEX, the computational system C_{HL} and the interaction between the two. But we need to say something about (3i), namely the notion of information to the interfaces—at least in order to specify the concept Exclusivity further. I will thus discuss the notion of address (identification) and how it fits in, before comparing this to other, similar approaches.

² See for example the work done in Distributed Morphology on some of these problems (Halle & Marantz 1993 and much subsequent work).

A simple of view of the interaction between C_{HL} and the interfaces (qua the intermediaries LF and PF) maps all information accumulated in the course of the derivation and maps it onto the interface levels LF and PF at a particular point in the derivation. This point is often referred to as Spell Out, the point of moving from overt to covert syntax. In the original minimalist model of Chomsky (1995b), Spell Out is a unique operation that targets the derivation at one point—roughly the same that was previously held to constitute the boundary of S-structure, but without forcing any conditions on the derivation up to that point. In order to keep the complexity of C_{HL} as simple as possible, it applies only once, apparently because this is the most economical way. However, if economy is really taken to mean “up to convergence,” Spell Out should be applied more than once if it can be shown that some derivation would not converge had Spell Out not applied more than once (Uriagereka 1998: 322). Uriagereka (1999b) does this and suggests that Spell Out should be treated like any other operation of C_{HL} , apply as often as needed to converge. In this context, the question arises how often Spell Out should apply. The answer Uriagereka gives is once per command unit, basically, every time a left branch is formed. Chomsky (1998, 1999) picks up on the idea of Multiple Spell Out and partitions the clause into phases, parts of the derivation that correspond to propositional objects which converge, where convergence can be taken to denote an application of Spell Out, a point in the derivation at which syntactic information is sent to the interfaces.

We will come back to both approaches soon, but for now take the idea seriously that the operation Spell Out can be applied more than once and that the interfaces are thus not computed over the entire structure—in other words, at least the consequence that LF and PF are fed cyclically can be upheld, regardless of the details of “Spell Out.” Given the (technical) discrepancy in this respect between Chomsky’s and Uriagereka’s models, it is not implausible that another way of implementing the idea of Multiple Spell Out can be envisioned. Leaving a deeper discussion aside for the purpose of presentation, let us consider its relevance to Prolific Domains and the framework developed here.

I suggest that (3i) be understood as follows. Once all features inside a collection of projections of the same type are checked, the derivation up to this point is evaluated. This collection refers, of course, to a Prolific Domain, as just sketched and evaluating thus concerns each Prolific Domain in turn (or cyclically). Evaluation consists of sending the information to LF and PF. Let us consider this in some detail.

If Spell Out is indeed the operation that ships the relevant information to PF and LF, and if it can apply more than once, PF and LF should not be considered to be one-dimensional levels but, what Uriagereka (1999b) or Uriagereka & Martin (1999) call, “components”—collection of information that is computed cyclically, one set at PF and another at LF. Such a cyclic application of Spell Out then takes each Prolific Domain as the relevant sub-part of the derivation that needs to be spelled out, or as I want to put it, evaluated. Evaluating a Prolific Domain consists of sending each object in the relevant part of the phrase marker to the PF- and LF-components. As many of these objects occur in more than one Prolific Domain in the course of a given derivation, it is necessary to mark them in a uniform way throughout. Call this the “address” of an object in a phrase marker.³ Each address of, or Spell Out point for, a given object, marks the LF- and PF-features of that object at that point. Arguably, the notion of address is important for LF as it identifies all features of one element relevant for interpretation, so that at the end of the derivation the entire set of checked features—for example, {[agent], [3rd person singular masculine], [accusative], [Wh]} for *who* in *Who did John kiss?*—can be matched to all occurrences of *who*. On the PF side this is presumably easier, as each occurrence looks alike, in this case [hu:] spelling out the θ -domain, [hu:] spelling out the ϕ -domain, and [hu:] spelling out the ω -domain.

³ This term has been used by other scholars in slightly different ways, yet always as marking a special occurrence. Manzini (1992) and Uriagereka (1997), for example, tie this special occurrence “address” to Case-marking, relevant for a theory locality in one case and an optimal, economy-driven checking approach in the other, respectively.

This is a simplified view, of course, but one that will suffice for presentation.⁴ We can think of an occurrence of an object as a specially coloured element. All non-distinct copies are of the same and each different lexical item of a different colour. A simple derivation for *John kissed Mary* contains roughly the following LA:

- (4) LA = {John, kissed, Mary, v, Agr, T}

Let us concentrate on the maximal projections here, the DP *John* and the DP *Mary*. At the end of the derivation, both will have checked their θ - and ϕ -features, and both have different Cases. Assume that each Prolific Domain is a Spell Out or evaluation point, so that after overt syntax, we get the simplified representation in (5) or shown in tree representation in (6) below.⁵

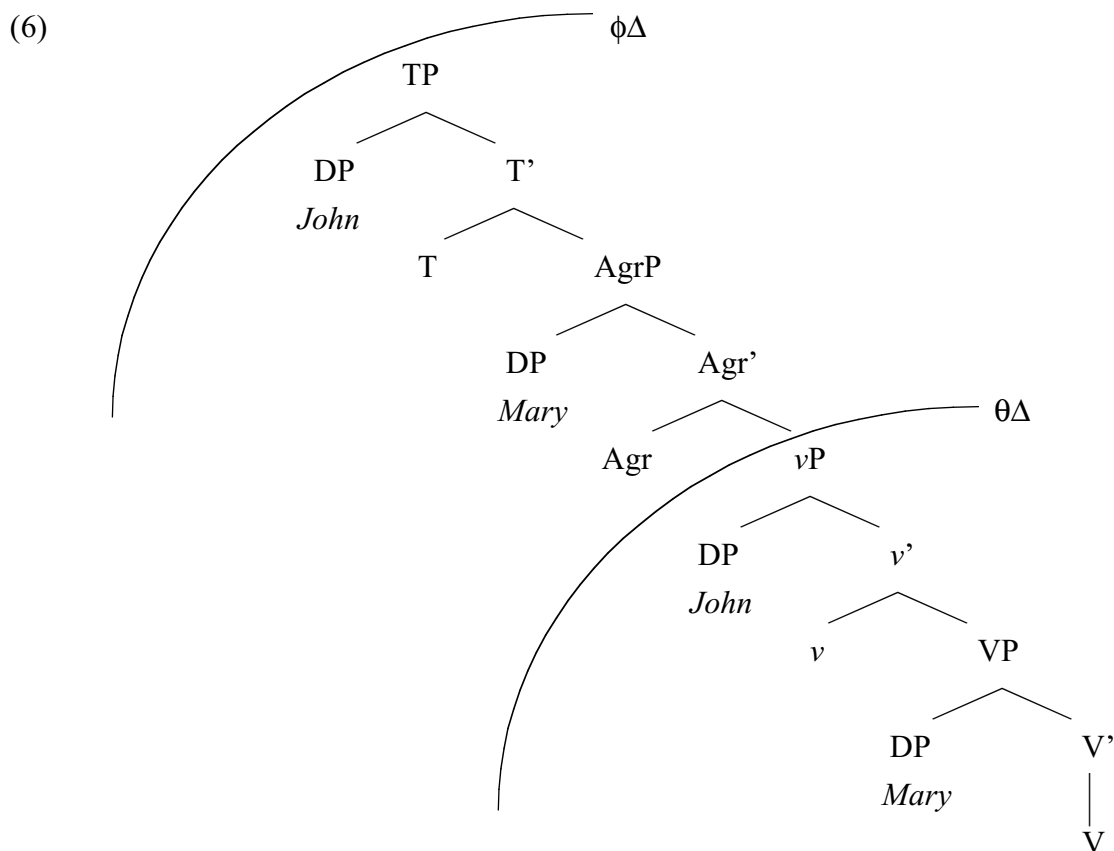
- (5) [TP John T [AgrP Mary Agr [_{vP} ~~John~~ kissed-v [_{vP} kissed-V ~~Mary~~]]]]

Leaving further details aside, the two relevant maximal projections, DP_{Mary} and DP_{John}, are spelled out first once the θ -domain is created, namely when vP is complete. The PF-component receives [dʒɔn] and [mɛərɪ]. Under a Single Spell Out model, LF would be created at the end and would only have to count all occurrences of the same item and take the relevant features checked at each occurrence. Under a Multiple Spell Out view, we have to say something else. Presumably, all that is shipped to LF at the point of θ -Spell Out is something like [theme] for *Mary* and [agent] for *John*, the two θ -roles. Both DPs still have at least their ϕ -features to check, and there is no need to inform

⁴ Recall that under the minimalist view, insertion of lexical items from the LA to the derivation includes the entire PF-matrix of the element as it is eventually pronounced.

⁵ To illustrate the point, assume all DPs move overtly, whether this is English or German, under our set of assumptions, for example.

LF that they have not yet been checked. At the next point of Spell Out, when the ϕ -domain is created, the two DPs are again shipped to the PF-component in their output forms [dʒɔn] and [mɛəri]. PF now has two of these and must decide which one to delete and which one to keep. LF receives the information [3rd person singular feminine, accusative] and [3rd person singular masculine, nominative] for *Mary* and *John*, respectively. However, it does not receive anything else; LF now has to match these features with the features already accumulated, such as [theme] and [agent].



This might be a crude sketch of the interaction between syntactic computation and the interface components, and it begs for more details. However, it suffices to illustrate the less ambitious goal of the current work to lay out a theory of anti-locality. We now have a way of understanding Copy Spell Out in terms of a PF-requirement.

By calling an occurrence of an object in the phrase marker “address” and encode the information shipped to LF as “address identification” (or Address Identification, capitalized by convention), LF is able to match [3rd person singular feminine, accusative] with [theme] and [3rd person singular masculine, nominative] with goal, regardless of the PF-structure of these two objects. The matching is done by Address Identification, which we could think of as different colours. The first of these two matched sets would be red, the second blue, for example. In this sense, Address Identification is nothing but a book-keeping device for the LF-component.⁶

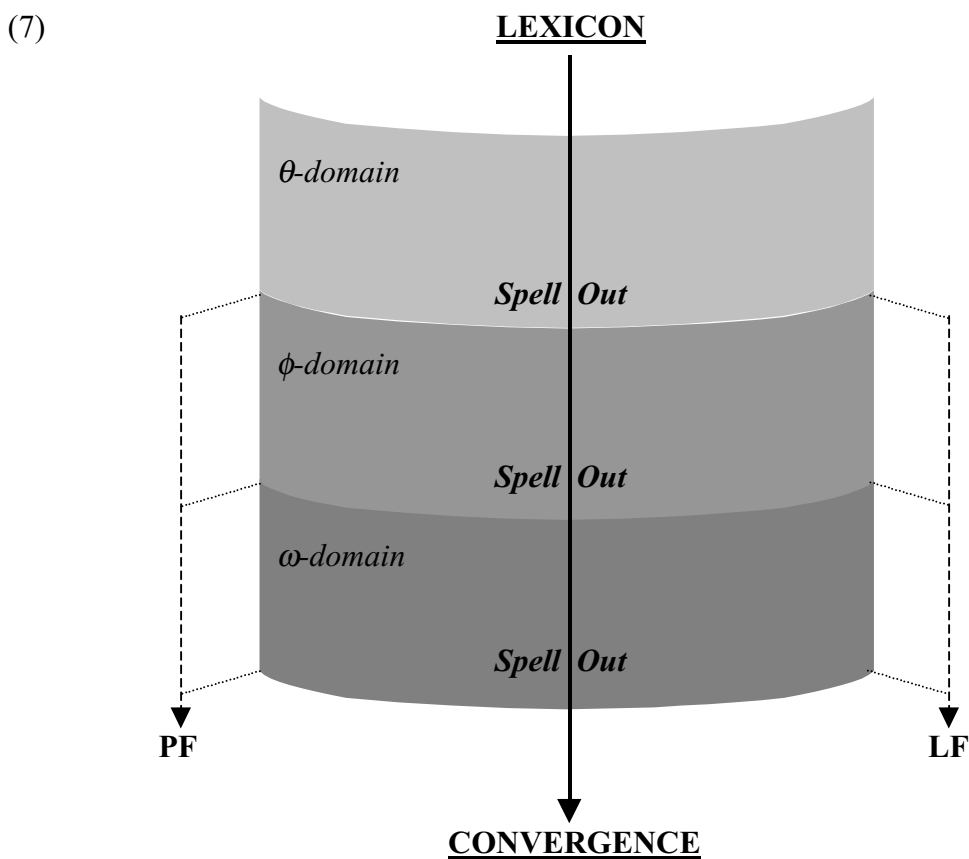
Exclusivity is defined in terms of a unique Address Identification, which is a uniquely coded (set of) feature(s) or an occurrence at LF. If coloured sets of LF-features indicate identity of sorts (in the form of matching, as described above), it should not matter whether there are two different-coloured features of a particular context (such as [agent] and [theme]) or two identically coloured ones. In the latter case, the two would be matched in relation to each other, and, depending on further content of these features, *John likes himself* or *John and Mary like each other* are potential results. For PF, on the other hand, the difference between *John* and *Mary* vs. *himself* is quite important: in one case, there are two different items, each of which needs to be pronounced once, while in the other—under the derivational approach sketched in chapter 3—is a copy that should not be deleted as usual, but pronounced. The repair strategy is to spell out one of the two copies bearing these features and recolouring it for PF purposes. Thus, if the LA in (4) does not contain *Mary*, the “red” DP *John* would move from one θ - to another θ -position, and there would be two red occurrences at Spell Out. But by previously spelling out the lower occurrence, marking the PF-matrix blue, the two occurrences are made distinct, PF

⁶ I abstract away from further complications, in particular the distinction of interpretable and uninterpretable features, an innovation which might be dispensed with (which would then be one of the more ambitious goals of this framework). This is just a sketch, with the aim to clarify the idea behind Exclusivity. I leave further details open for future research.

does not have to delete any copy, and the LF-component can happily match the relevant features accumulated in subsequent operations (under identity of sorts, i.e. it only deals with red features throughout the derivation for both instances of *John*).

Apart from the issue raised in fn. 6, this view implies two things. First, Copy Spell Out must take place before the relevant Prolific Domain is established, otherwise LF would receive conflicting information (by (21)). Second, a Prolific Domain is established or evaluated (Spell Out) as soon as it is complete, thus allowing an element to move beyond. This will be addressed more in the next section.

If this is on the right track, we might want to modify the model of the grammar shown in (8) in chapter 1 along the following, more colourful—if only tentative—lines (from Grohmann 2000a: 42), regardless of whether we take Spell Out to be the relevant notion of something less committal (“evaluation”):



To conclude our review of the Domain-driven framework presented here, we now have a way of conceptualizing the assumptions on the computational system underlying the definitions of Prolific Domains and Exclusivity. By evaluating locally—that is, at the point of establishing—each Prolific Domain, an occurrence of an object in a phrase marker must be unique, or exclusive, at the point of evaluation, which is within a given Prolific Domain. Spelling out the lower copy can repair a potential violation, but must take place before evaluation.

Thinking of occurrences in colour terms is one way of looking at things. Another analogy would be with the realm of quantum physics. Exclusivity as viewed here is in a way reminiscent of Pauli’s Exclusion Principle, which says that “no two electrons in an atom can have the same set of quantum numbers” (Serway, Moses & Moyer 1989: 230, emphasis omitted—KKG). What the quantum numbers do is identify the position of an electron within an atom. What feature checking does is identify the position of an object in a phrase marker, in particular within a given Prolific Domain. There are many more examples from other areas one could draw from, but the bottom line is that Exclusivity of an element within a specially contained area seems to be a natural state of affairs.

6.3 Exclusivity, (Anti-)Locality and Cyclicity

Aside from many other consequences, this sketch allows us to compare the present system with other, similar ideas on a dynamic conception of syntactic computation, such as the theory of Multiple Spell Out (Uriagereka 1999b) and the (re)introduction of phases (Chomsky 1998, 1999). The main advantage of the first is that it derives the LCA immediately, on the assumption that all command units (basically, left branches) are spelled out, thus linearizing all objects in the phrase marker. Uriagereka treats the opera-

tion Spell Out to basically “morphologize” every object, turning it into one word at the interfaces, and as a consequence prohibiting extraction out of a spelled out element. This might be the right way to understand Spell Out, but it is different from the purpose ascribed to it here.⁷ As such, Multiple Spell Out and Prolific Domains are not necessarily mutually exclusive. Perhaps a better way of characterizing the above sketch is not in terms of spelling out, but in terms of evaluating. The PF and LF components thus evaluate every Prolific Domain locally, at the point where it is created, leaving open further specification of the operation Spell Out, perhaps in Uriagereka’s terms.

The idea of a phase, however, is not as simple to integrate. In the following, I pick out the core properties Chomsky (1998, 1999) ascribes to a phase and the correlates found with Prolific Domains.⁸ The upshot will be that the framework of Prolific Domain fares no worse than a phase-based system in conception or empirical coverage.

For the purpose of a brief summary, Chomsky suggests that vP and CP induce (strong) phases, essentially bounding nodes for subsequent movement. An element can only move out of a phase into a higher one from an edge position. It is obvious, then, that a phase cannot correspond to a Prolific Domain. First, we treat TP as a “phase” as well, and second, the structure that allows an element to get to the edge of a phase and then move further is banned from the system endorsed here. The edge of a phase is basically the relevant head (v , C) and *all* of its specifiers. In the current framework specifiers are

⁷ It also faces an empirical challenge in that extraction out of left branches is not prohibited in all languages (such as *was für*-split in German, for example, which is insensitive whether it applies to a subject or an object), but this implementation presumably depends on other factors, such as the ultimate structural analysis of the “base structures” of these strings. See also Stepanov (2000) for discussion and references.

⁸ Note that Chomsky (1998) introduces two phases, basically vP and CP . In Chomsky (1999), he distinguishes weak from strong phases. As far as I can see, the relevant notion for our present comparison is a strong phase (vP , CP), while weak phases shall not concern us here. (One weak phase is a VP , or deficient vP , which is relevant for the allegedly phonological operation THematization/EXtraction, for example, in which case TH/EX alone may “spell out” rather than the entire phase.)

unique, hence no element can use an outer specifier as an escape hatch. The assumptions on feature checking are also slightly different, as a consequence of multiple specifiers on the one hand (where the object checks its ϕ - and Case-features against v) and the option to check features without movement on the other (via Agree). Let us view these and other properties of phases in a little bit more detail.

Chomsky (1998: 21) defines a phase as “propositional” and “convergent.” The latter relates to the cyclic operation of Spell Out/evaluation and could translate as such into the Domain-driven framework: Prolific Domains and phases are convergent, they both induce Spell Out. Regarding the former property, it is not quite clear what the details of a “propositional” property of phases are (see e.g. Uriagereka & Martin 1999, Pietroski 2000, Uriagereka 2000, or Juan Uriagereka’s University of Maryland fall 2000 lectures). I suppose vP could relate to the (minimal) event structure of the clause, while CP would be propositional in that it encodes all other specifications and basically completes the clause. I would like to contrast this vague understanding of “propositional” (coupled with the question mark why TP does not induce a phase, at least a non-deficient TP) with our property of a Prolific Domain, namely that it is essentially “contextual.” I have offered suggestions in how far a Prolific Domain is “contextual” in chapter 2, and we have seen further evidence throughout. In the two core properties of definition, Prolific Domains and phases seem to fare equally well or badly. Let us now look at some technical aspects.

A phase-driven framework is subject to the Phase Impenetrability Condition (or PIC; Chomsky 1998: 22, 1999: 10), which basically states that elements within a phase are not accessible to further computation at the next higher phase, unless they are located in the edge of that phase. The Domain-driven framework takes the opposite stance, namely that an element cannot move within the same Prolific Domain (Exclusivity qua the definition of the CDE in (2)). That PIC and CDE can indeed be compared in this context follows from the following divergent assumptions on the syntactic derivation between the two frameworks.

First, we take displacement to be triggered by a property (“feature”) on the displaced element itself, that is we extend the original conception of Move (Chomsky 1993). The phase-driven approach assumes the trigger to lie in the target (“Attract”). Moreover, much of the motivation that led to movement in earlier minimalism is now dispensed with and uninterpretable features are eliminated in a long-distance configuration by the operation “Agree.” This is as far as we can compare the basic background assumptions. A more detailed evaluation of Move vs. Attract/Agree is beyond the scope of this work and depends on many other assumptions (see, for example, fn. 6).

Second, we eliminated the possibility of multiple specifiers, or anything more than one maximal phrase per projection created by movement. As such, an edge can never be created in Chomsky’s sense, as an escape hatch. These two differences account for the opposite framework that the PIC and the CDE yield, on the one hand, and for the similar spirit the two imply, on the other. The comparison of differences and similarities thus extends further. Recall that we take the position that a Prolific Domain is evaluated as soon as it is created, while the relevant equivalent of Spell Out in a phase-driven framework takes place at a later stage: a lower phase PH1 is only then spelled out and made inaccessible for further computation once the next higher phase PH2 is reached. This allows elements in PH1 to move to the edge of H1 (the head of that phase) and be attracted by H2 (the head of the higher phase). Thus, late Spell Out of phases allows for an escape hatch for material that needs to be displaced, while early Spell Out of Prolific Domains freezes all Domain-internal movement.

This is a rough comparison of the theoretical underpinnings of the two theories. Let us then look at empirical coverage. In order to move out of a phase, an element must first move to the edge of that phase. This is not feasible in a Domain-driven approach and the most natural counterpart of movement to the edge is movement to the next Prolific Domain. For standard cases, this seems to make the right cut. Thus, an outer Spec ν P checks those features that SpecAgrP checks here, namely ϕ - (and Case) features.

We could reinterpret this by suggesting that the vP -phase and the θ -domain are the same, if the former is cut off after its first specifier, the thematic subject position. Regardless of numbers of specifiers, anything beyond would then fall into a different phase or Domain. But this easy transposition cannot be upheld because TP (or the ϕ -domain) does not constitute a phase in the relevant sense. In all simple, monoclausal structures this difference does not amount to much. Take Wh-movement, for example. In both frameworks a direct object Wh-phrase is generated in its θ -position and ends up in the surface Wh-position (specifier of FocP or CP). In one framework, it moves through AgrP, the ϕ -position, for by now obvious reasons, then targets the final position inside the ω -domain, that is, it moves from the θ - to the ϕ - to the ω -domain. In the other, it would be frozen in its base position once we reach “C” (the higher phase), unless it moves to the edge of the lower vP -phase, from where it is accessible to subsequent movement to SpecCP. I take it that both approaches embody the same necessary assumption: that all movements must target the next highest Prolific Domain for principled reasons (to compute a complete set of context features) or that all movements must use the escape hatch at the edge (which Chomsky calls “independent feature-driven movement” or IFM; Chomsky 1998: 21f.). I do not want to question the “naturalness” of either alternative, but consider some cases which can be compared, which arguably suggest that only vP and CP are phases, and which empirically support the phase-driven approach within the relevant set of assumptions. These cases involve clause boundaries, or multiclausal structures.

In Chomsky’s framework, movement from one thematic position to another is ruled out, and control structures contain a controlled empty subject in the embedded infinitival clause, PRO. In the current framework, we do not want to exclude θ -to- θ -movement a priori and dispense with PRO as an object in C_{HL} . Adapting Hornstein’s (1998, 1999, 2000) analysis of control constructions slightly, along the lines suggested by Castillo, Drury & Grohmann (1999), we take the controlled subject to be a copy of the phonetically realized subject. Compare the two derivations:

- (8) a. [TP John T [_{VP} ~~John~~ tries-V-*v* [_{CP} C [_{TP} PRO to-T [_{VP} (~~PRO~~) sleep-V-*v*]]]]]
 b. [TP John T [_{VP} ~~John~~ tries-V-*v* [_{TP} to-T [_{VP} ~~John~~ sleep-V-*v*]]]]

The controlled embedded infinitival in (8a) needs to be contained in its own phase, otherwise anything beyond the *v*P-phase would be accessible to the matrix *v*P-phase.⁹ This can be done by postulating CP in between the two clauses (but see fn. 9). (8b) does not need to stipulate the presence of CP, as it does not need to induce a phase. Moreover, if the controlled subject is indeed the copy of the matrix subject, we do not need to treat the embedded grammatical subject position as present. If non-finite T cannot license a phonetically realized element in these contexts, there is no reason to assume it in the first place—unless one would want to express the Extended Projection Principle (EPP) somehow. But this has come under attack in recent work.¹⁰ This allows for a generalization on locality conditions in a Domain-driven framework:

- (9) *Domain-driven Locality*
 a. Intra-clausal movement targets the next higher Prolific Domain.
 b. Inter-clausal movement targets the same type of Prolific Domain.

We have seen instances of (9a) throughout, and the ban on the opposite, with the caveat of spelling out lower copies (CDE). The derivation in (8b) suggests something like (9b), and while a complete demonstration lies outside the scope of the current paper, some core cases can be illustrated, suggesting that (9) is indeed on the right track.

⁹ The structure portrayed in (8a) is one potential derivational history for this sentence, where it does not matter for current purposes whether PRO is inserted or derived. What is relevant is the fact that CP is forced, for phase-reasons, despite evidence that control clauses involve a TP, as noted by Uriagereka & Martin (1999); see Ormazabal (1995), Bošković (1997c).

¹⁰ See, for example, Castillo, Drury & Grohmann (1997, 1999), Epstein & Seely (1999), Boeckx (2000a), Bošković (2000b), Grohmann, Drury & Castillo (2000), Nasu (2000), Lasnik (to appear), but also discussions in, among others, Boeckx (1998), Kidwai (2000), Roberts & Roussou (2000).

Take the “counter-case” of control structures. The generalization in (9) makes one prediction, if correct: raising structures should contain intermediate SpecTP, on the assumption that a raising subject does not move through the matrix θ -position but targets the matrix SpecTP straight away (under the traditional approach) or that it needs to target the same type of Prolific Domain if it crosses a clausal boundary (current approach). The standard analysis of raising constructions is the following (Chomsky 1981, 1993):

(10) [TP John T [VP seems-V [TP ~~John~~ to-T [VP ~~John~~ sleep-V-v]]]]

The matrix subject is the grammatical subject of both clauses (traditionally, for reasons of the EPP) and the thematic subject of the embedded predicate only. Let us assume that the matrix clause does not have a ν P (see section 3.2.3). The relevant movement step of the matrix subject is the last one, from embedded SpecTP to matrix SpecTP. If this is indeed the right movement, it conforms to (9b): the move cross a clause boundary and targets the same type of Prolific Domain in the target clause as it originates from in the launching clause, here the ϕ -domain.

However, as we said above that the EPP should not be invoked, if possible, the intermediate landing site might be a thorn in our side. This is essentially the position Castillo, Drury & Grohmann (1999), and others, advocate: the EPP does not exist, hence there is no motivation for an intermediate touch-down in infinitival TP, hence non-finite (deficient) SpecTP does not exist. Thus the anti-EPP proponents mentioned above and elsewhere deny intermediate SpecTP in these constructions also. But these two sides, pro or anti EPP, are not necessarily the only options available. Bošković (2000b) proposes a reconciliation of both viewpoints, namely that the EPP does not exist,¹¹ but that certain

¹¹ As Lasnik (to appear) notes, the EPP has undergone considerable conceptual—and as a result, technical—changes over the past two decades. While treated as a formal feature in Chomsky (1993, 1995a), latest conceptions of Chomsky (1998, 1999) are much more in
(continued...)

intermediate subject positions do. One empirical case is the so-called *wager*-type verb class (Pesetsky 1991, Bošković 1997c). Verbs of this type select for a non-finite clausal complement whose subject position is filled by an expletive. To cut a long story short, if *there* can be inserted into intermediate subject positions in certain raising structures, that position should be available (but not in control structures). Bošković (2000b) compiles a wide range of data that suggest that both the derivations sketched in (8b) for control and in (10) for raising are correct.

This approach captures the best of a movement approach to control and an EPP-less theory, yet maintains a difference between the two other than in one additional θ -role. It also allows Uriagereka & Martin (1999) to keep (non-defective) TP as a phase, without, at the same time, getting into a conflict between the notions of Prolific Domain vs. phase, all other things being equal. Regarding the latter, the different analyses for control constructions allow phases and Prolific Domains to be empirically adequate in either framework, and we can take the decision between the two approaches to follow from the appropriateness of the background assumptions sketched here. Note that raising cases are handled equally well by either theory: the phase-driven framework does not want the matrix and embedded clauses be separated by a phase, while the Domain-driven approach captures the successive-cyclic nature of raising by (9b).

The following are abstract representations illustrating (9):

(11) *Intra-clausal movement*

$[\omega_{\Delta} \dots \text{XP} \dots [\phi_{\Delta} \dots \mathbf{XP} \dots [\theta_{\Delta} \dots \mathbf{XP} \dots]]]]$

(12) *Inter-clausal movement*

$[\omega_{\Delta} \text{XP} \dots [\phi_{\Delta} \mathbf{XP} \dots [\theta_{\Delta} \dots [\omega_{\Delta} \dots [\phi_{\Delta} \mathbf{XP} \dots [\theta_{\Delta} \dots [\omega_{\Delta} [\phi_{\Delta} \mathbf{XP} \dots [\theta_{\Delta} \mathbf{XP} \dots]]]]]]]]]]]]]]]]]]]]]]]]$

line with the original assumption that sentences need to have subjects, or basically, that certain specifiers need to be filled (Chomsky 1981, 1982).

(11) would apply to a sentence like (13a), while (12) would correspond to (13b):

- (13) a. Who did Mary ~~who~~ see ~~who~~?
 b. Who ~~who~~ seems ~~who~~ to be likely ~~who~~ to ~~who~~ sleep?

In a phase-driven system, the intermediate copy of *who* would sit at the edge of the relevant phase from which it can move to its final destination in (13a). In the current approach it sits in the next highest Prolific Domain (ϕ -domain) and targets another position in the next highest Prolific Domain (ω -domain). The relevant intermediate positions in (13b) can be identified with SpecTP and are driven by IFM in a phase-based theory and by the generalization on inter-clausal movement here.

6.4 Concluding Remarks

In this chapter I have reviewed the core proposal of the theory of anti-locality presented in this work. We now have a full-fledged framework that not only captures anti-locality, as we set out to do, but also offers a new look on the interaction between syntactic computation and the interaction with the PF and LF interfaces. A comparison with recent approaches aiming for the same goal yields the following picture. A cyclic application of the operation Spell Out is an important ingredient, whatever the exact details, and it needs to be seen what the correct portion of the derivation is which gets spelled out. We also saw how Prolific Domains and phases fare with respect to conceptual and empirical motivation. If nothing else, the ideas presented here make predictions like any other good theory and are thus testable (a desired result from a theoretical point of view). Independently, the analyses of empirical phenomena considered in the past chapters constitute a contribution to the analytical repertoire of minimalist research.