Tokens vs. Copies: Displacement Revisited

Diego Gabriel Krivochen

Abstract

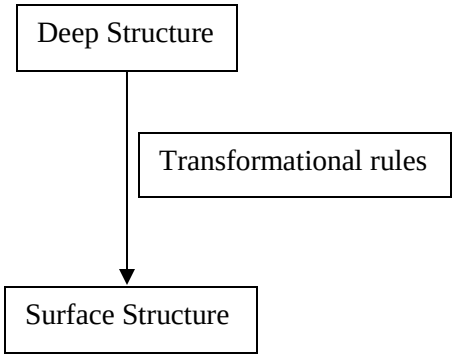
In this paper we will analyze the conceptual and computational motivations of the property of *displacement* in natural languages from a revisited perspective. We will account for *displacement* phenomena proposing our own version of *displacement-as-external token Merge*, as opposed to the traditional *displacement-as-literal movement* or, more recently, *displacement-as-copy and Merge* (Chomsky, 1995; Kitahara, 1997; Nunes, 2004). Our focus, as far as empirical data is concerned, will be set on parasitic gaps and their derivation, comparing our proposal with previous accounts making particular stress on the idea that operations are not feature-driven but interface-driven.

Keywords: Radical Minimalism, Movement, Syntax-Semantics interface, Free Merge, Type-Token dynamics

1. Introduction:

Natural languages, it is widely accepted, displays two fundamental properties: *hierarchical structure* and *displacement* (facts (v) and (vi) in Hornstein, Nunes & Grohmann, 2005: 7). Formal, explicit grammars have attempted to account for both in an elegant way, maintaining a balance between theoretical generality and abstraction and empirical adequacy. Chomskyan generative grammar, stemming from the seminal "Syntactic Structures" (1957), proposed a polystratal approximation to the problem, which has become mainstream and a clear identity mark of Chomskyan linguistics, to the point that "Generative Grammar" and "Transformational Grammar" were used as synonyms in the '70s and '80s. The architecture assumed in the Standard Theory (Chomsky, 1965) and the later GB model (Chomsky, 1981) can be schematized as follows:

1)



Transformational accounts (Chomsky, 1957, 1965, 1981, 1986), even if some are focused on the concept of isomorphism, depart from the assumption that there are (at least) two levels –or *strata*- of syntactic structure, a deep one (derived from the development of an L-grammar, see Lasnik, 2011) from which semantic interpretation is built, and a surface one from which phonological interpretation is built, linked by a set of transformational rules that apply to phrase markers in the deep structure and generate surface structures as output. The basic architecture has been modified in the different theories (e.g., in the Government and Binding model, the Surface Structure split the derivation in two, Phonetic Form and Logical Form), but the basic idea is always the same. Transformational models are *obligatorily* polystratal theories (Culicover & Jackendoff, 2005), as they are forced to assume more than one level of structure and a derivational relation between the representations generated in each of them. Moreover, a set of linking rules is also necessary to relate both structural descriptions as deriving one from the other, particularly if a system with interpretative interfaces is assumed. Our goal will be to eliminate as much from this machinery as possible.

1.1 The Standard Theory:

The early architectures (1957, 1965) assumed a *base* component including both a LEX and Phrase Structure Rules (PSR) that strongly generated structural descriptions of sentences. PSR adopted the form of an L-grammar, $X \rightarrow Y$, a format already known in phonology to describe allophony, with the

only difference that PSR did not necessarily need contextual specifications. A different set of rules, Transformational Rules (TR) applied to the initial Structural Description (SD) and generated a Structural Change (SC), with displaced constituents. Let us give an example:

2) SD: $NP_1 + V + NP_2$

   TR: passive (optional)

   SC: NP_2 + be + V-ed (+ by NP_1)

As the reader can see, in the SC there is no trace of the operation that applied, nor is there any principle that impedes us to say that SC is actually the kernel sentence generated by PSR: this is a crucial point for Emonds' (1970) *Structure Preserving Hypothesis*: a rule or rule system is "structure preserving" if and only if its output is independently available as an underlying structure, in this case, a SD. Those dependencies, in the case of optional transformations, were assumed departing from traditional considerations: passive derives from active, being both well-formed formulae. Not quite the same was the case with obligatory transformations, like "do-insertion" in interrogatives, which assume no derivational relation between SD and SC as two well-formed formulae, since if a transformation is obligatory, it is so to "save" the representation, to make it well-formed. It is indeed curious why the system requires the whole package of SD, TR and SC in the case of obligatory transformations and not directly generate SC as the SD in the first place. A possible answer lies on PSR, which stipulate a particular instantiation of an L-grammar. But take, for instance, (3) (adapted from Chomsky, 1957: 112):

3) SD: NP – V – Prt – Prn

   TR: T_{sep}

   SC: NP – V – Prn – Prt

For example:

4) a. SD: *The police brought in him (Chomsky's example (83))

b. SC: The police brought him in

It is curious why the base component would not be able to generate (4 b), but has to derive it via movement instead. That is, there is no PSR stipulating that VP → V – Prt, which would make (4 a) the basic kernel. The status (nature and ontology) of the transformational component was dubious (and it remained so in the early 70's, with the generative semantics conflict and the opposition "all-powerful transformational component vs. all-powerful base component". See McCawley, 1968a, b and Chomsky, 1972 respectively), but the status of the base component was not any clearer.

1.2 Government and Binding model:

This situation changed in the late 70's, particularly after "Conditions on Transformations" (Chomsky, 1973) and, mainly, "Filters and Control" (Chomsky and Lasnik 1977). The simplification of the theory and its universal validity were the main concern now, and very language-specific rules were undermining explanatory adequacy. The 80's witnessed the development of a new conception of displacement: *Movement*. In the "Lectures on Government and Binding" (Chomsky 1981) framework, grammatical constructions were (allegedly) eliminated as primitives of syntactic theory, and PSR were simplified to a single requirement: *Satisfy* (lexical requirements of predicates) forming a level of representation known as D-Structure, in conformity with X-bar theory, a module of the grammar which shared the L-grammar format of PSR but not quite their purely derivational character. The model drifted towards representationality. Syntax was now a projection of the lexicon's c- and s-requirements (see also Stowell, 1981 for important developments as far as phrase structure is concerned). TR also underwent important changes, to the point of being subsumed to a single algorithm (even though it was never formalized as such) Move-α, that is, "move anything anywhere"

(while respecting Principles restricting Movement, see Lasnik & Saito, 1992 for details). The clues this operation left were so-called "traces" in the base-generation place of displaced constituents (be them arguments or adjuncts), as in the following example:

5) What$_i$ did Mary buy t_i

The Wh-word generates as the complement of V, being its internal argument. There, it receives a theta-role and Case via government by V. When it moves to the Specifier position of the C head, it leaves behind a trace that is to be interpreted at Logical Form and erased at Phonological Form. The visible consequences of this application of Move-α are to be seen in S-Structure, a level of representation that is the result of applying transformational operations to D-Structure, which result in a new phrase-marker (P-marker). Great part of the work in GB was devoted to the property of displacement (since hierarchy had been apparently taken good care of via X-bar theory) and inter-linguistic variation with respect to extraction and movement possibilities: thus, the "parametrical approach" to acquisition is to a considerable extent a set of conditions over movement possibilities. Consider, for example, that the so-called *pro-drop parameter* (Jaeggli & Safir, 1989) has consequences for extraction across an overt C, as Haegeman & Gueron (1999: 599, ff.) explain. Parameters were not associated with a single property, but most often with a cluster of properties of a given language L, many of which were related to Movement possibilities. Consequently, a theory of unpronounced traces had to be developed, in order to formalize not only the possible distribution of traces but also the relations with the displaced constituents, which gave the trace reference and with which they shared either Case or Theta-Role. Thus, empty categories, a label that included both traces and null subjects of finite and non-finite clauses, were paired with overt elements in syntactic representations using two binding features, [± pronominal] / [± anaphoric]. The resulting chart has the following form:

[a]	[p]	Symbol	Name of empty category	Corresponding overt noun type
-	-	*t*	Wh-trace	Referential expression
-	+	*pro*	"little pro"	Pronoun
+	-	*t*	NP-trace	Anaphor
+	+	PRO	"big pro"	????

Table 1

This featural characterization translates in the following distributional constraints:

6) Wh-traces and *pro* are never correferential with a c-commanding element within a local domain D
7) NP-traces and PRO can be correferential with a c-commanding element within a local domain D

The ruling principle was the so-called Empty Category Principle, formulated as follows:

8) *All traces must be properly governed*

Where "proper government" is understood as in (9):

9) *e* [an empty category] is properly governed by A iff A c-commands *e* and A is a lexical item (that is to say, A ≠ AGR (Chomsky 1981: 250))

In turn, proper government can be:

10) a. lexical government (the trace is c-commanded by a lexical head that assigns it a θ-role)

b. antecedent government (the trace is c-commanded locally by its antecedent, and there is no barrier between them)

Basically, a trace must be c-commanded by its antecedent (and, for that to happen, movement should always be bottom-up, whether in the syntax proper or in LF). Reference (i.e., coindexation), for example, is always interpreted in a left-to-right fashion at the LF interface (that is, in unmarked structures, the element with "absolute reference", say, an R-expression or a finite V form is always c-commanding that with "relative reference", adopting and adapting the labels from Tense relations), as was captured in binding principles A and B. From the conditions outlined above, we can deduce that moved adjuncts must be antecedent governed (as they are not θ-marked), and moved arguments (being θ-marked) must satisfy both conditions. Conditions on the relation between trace and antecedent, as we see, are constrained by the notion of c-command, in turn parasitic on phrase structure. Since phrase structure is, in turn, parasitic on lexical specifications (via the Projection Principle), the situation is not very different from that we would find in alternative non-transformational models like HPSG, with the only difference that there is no D-/S-Structure dynamics but the empty slot(s) in the structure is specified in the lexical entry of the predicate licensing the position (Green, 2011: 38, ff.) in the form of a SLASH feature. In any case, HPSG also uses the term "extraction", and, while eliminating purely syntactic constraints in favor of a theory of "competence", throws all the responsibility to pragmatics[1], an incorrect move in our opinion as there is no articulate and explicit pragmatic theory to truly account for gap phenomena, at least not yet. However, we do agree with HPSG (and pursue this venue) in that

[1] "(…) *any constraints on them* [gaps] *are pragmatic, not syntactic, in nature. In fact, the HPSG treatment of gaps predicts that in the general case "extractions" from the clausal complement of nouns will be syntactically well-formed* (…)" Green (2011: 41). Also, Levine & Sag (2003: 6): "(…) *there is now increasing evidence that a good number of these* [island] *phenomena are due not to constraints of grammar, but rather to processing, pragmatic, and discourse effects that can be manipulated in such a way as to ameliorate the severe unacceptability exhibited by* [examples involving violations of island constraints]".

an interface theory is needed to replace intra-theoretical stipulations about syntactic features affecting gaps, and the desire to unify all gap phenomena (as we will see in Section 3.1).

1.3 Minimalism and beyond:

The advent of the Minimalist Program in the early 90's brought stricter economy considerations to the theory, particularly the elimination of D- and S- structures via the subsumption of the principles that applied at those levels to interface conditions and the redundancies that were found between the levels themselves (for example, S-Structure contained the very same information as D-Structure, as traces and indexes do not add information) and the *Inclusiveness Condition*, that banned the introduction of new elements in a syntactic derivation, that is, elements that were not present in the Numeration (a subset of the Lexicon used to derive a particular sentence). To this end, Chomsky (1995, Chapter 3) introduced the first version of the so-called Copy Theory of Movement (CTM), according to which displacement keeps being interpreted as movement, but implemented as a composite operation (see also Kitahara, 1997: 7, ff.):

11) Who did you see?[2]

12) *Copy* the relevant constituent: [who] [C you did see who]

    Merge the relevant constituent (respecting the Extension Condition): [[who] [you did you did see who]]

    Delete lower copies for PF purposes: [who did you ~~did~~ see ~~who~~]

Since Chomsky does not formalize (or even mention) the notion of workspace (which only came to the front line in generative linguistic theory with Uriagereka's 1999 paper on derivational cascades

[2] We will not get into do-insertion, because there is no satisfactory theory of do-insertion within orthodox minimalist assumptions. Within Optimality Theory, Grimshaw (1997) is the classic study of do-insertion as a last-resort to satisfy constraints, but the "solution" is just as unprincipled as any of those proposed within orthodox minimalism under the "last-resort" label, particularly those being parasitic on the "V-to-T movement" approach.

and Multiple Spell-Out), it is difficult to picture how this would work, since the "copy" must exist somewhere before being Merged. That, naturally, assuming there is some kind of diachrony to the operation: if Move is really complex, then these must be *steps*, and there is a step, Copy, in which a constituent must be maintained in some kind of *short time memory* until Merged, but the copy theory proposes no such procedure. This is most striking, since Chomsky himself has been one of the most enthusiastic proponents of so-called "biolinguistics", and an explicit neuro-cognitive model where to implement the computational machinery would be most expected. Later refinements of the CTM addressed the issue of Movement motivation in different terms with respect to early Minimalism: whereas in the Chomsky (1993) system Last Resort and Procrastinate were still valid principles and the role of uninterpretable features in syntactic derivations was not yet developed, Chomsky (1999, 2000) presents a much more complicated picture, in which the basic claim "Merge-over-Move" is maintained on the grounds of "good design conditions" (kept unformulated to this day, see the criticism in Behme, 2012), but the details of implementation greatly change. The reasoning is as follows:

> *"UG makes available a set F of features (linguistic properties) and operations* C_{HL} *(the computational procedure for human language) that access F to generate expressions. The language L maps F to a particular set of expressions Exp."* (Chomsky, 2000: 100)

Furthermore, those "linguistic properties" come in four varieties: interpretable, uninterpretable, valued, and unvalued (Chomsky, 1995: 229-30; 1999: 8), depending on the category these features compose[3]. Correlations between these varieties are not agreed upon (confront Chomsky, 1999: 10 with Pesetsky & Torrego, 2004: 2-5, for instance), but the properties are widely accepted. Now, uninterpretable features have to be eliminated from the derivation that maps NUM to PF and LF. The

[3] For example, φ-features are interpretable and valued in N/D, but uninterpretable and unvalued in T.

way to eliminate uninterpretable features is via the operation Agree (and concomitant valuation, depending on the presentation), which

> "(…) *establishes a relation (agreement, Case checking) between an LI α and a feature F in some restricted search space (its domain). Unlike Merge, this operation is language-specific, never built into special-purpose symbolic systems and apparently without significant analogue elsewhere. We are therefore led to speculate that it relates to the design conditions for human language.*" (Chomsky, 2000: 101)

Provided that Chomsky does not even consider the possibility that the "far-fetched" character of *Agree* may be given by the fact that it plainly does not exist (that is, that features, if existent at all, have nothing to do with displacement), he keeps arguing:

> "*A third operation is Move, combining Merge and Agree. The operation Move establishes agreement between a and F and merges P(F) to αP, where P(F) is a phrase determined by F (perhaps but not necessarily its maximal projection) and αP is a projection headed by a. P(F) becomes the specifier (Spec) of α ([Spec, α]).* […] *Plainly Move is more complex than its subcomponents Merge and Agree, or even the combination of the two, since it involves the extra step of determining P(F) (generalized "pied-piping")*". (Chomsky, 2000: 101)

To summarize, Agree, which is in turn composed of *Search* and *Valuate*, or *Agree* (quite a confusing terminology, indeed), an operation applying to individual features, is itself part of Move, which applies to bundles of features. In later developments (Chomsky, 2004, 2005, 2007), the Probe-Goal relation involved in *Search* was enriched with the condition that, for *x* to internally Merge to αP, the head of αP (say, α, under standard assumptions), had to be endowed with an EPP feature, triggering Merge to the periphery of the relevant head, thus assuring the fulfilling of the Extension Condition:

any operation must extend the root tree, with the additional claim that no operation can internally change any of the objects involved (the so-called No Tampering Condition. Chomsky, 2005: 5).

After reviewing the most influential concepts of displacement within generative grammar, we will devote the rest of the paper to presenting our own proposal, grounded on the assumption that *displacement does not equal literal movement*, and implementing a free derivational engine only constrained by interface conditions, among which –we will claim- semantic conditions are preeminent.

2. Move as Token-Merge

We have seen that the inclusion of traces in a derivation violates the Inclusiveness Condition, and therefore, they were replaced by Copies. However, the operation Copy also introduces new elements in a derivation, provided that the copies are not present in the NUM. Conversely, if they were, the operation Copy would be superfluous, since there would be nothing to copy. In any case, the problems arise not only at the syntactic workspace, regarding the motivation for displacement and phonetic realization of copies (ranging from deletion of uninterpretable features and c-command relations to alleged PF "principles", see Chomsky, 2000 et. seq.; Nunes, 2004; Larson & Hornstein, 2012 for fine examples of the former approach, and Kayne, 1994; Bošković and Nunes, 2007, and Moro, 2000 for developments within the latter), but also at the C-I interface for reconstruction purposes: how does the interpretative system it is dealing with more than one instance of the same object? And, at which level is it the same object, anyway? Some of these problems have been addressed by the CTM and, more than anything, Multidominance theories of movement (MTM) (see Vicente, 2009 for a comparison of both approaches, and Larson & Hornstein, 2012 for a very quick –and possibly shallow- overview), but there is still an unsolved question: assume that, as Chomsky proposes, Move is actually Merge. We have to know where the merged element comes from: either the NUM or the derivation. Both alternatives, as traditionally conceived, have their disadvantages: on the one hand, re-merge from the derivation requires that both the target and the displaced constituent be in the syntactic workspace at

the same derivational point D_x. This means that, if some kind of cyclic computation model is to be applied (see, e.g., Bresnan, 1971, 1972; Uriagereka, 1999/2002; Chomsky, 1999), either displacement has to be a reason to procrastinate Transfer to the interpretative components or a version of successive cyclicity is to be implemented. Either way, the solution is stipulative (even though it can be wiped under the rug of "optimal design", as it has been): unless cyclicity is required by conceptual or phonological matters, it should have no place in a theory about the computational component. However, the introduction of feature valuation considerations made it possible to add Probes to which (copies of) displaced constituents could Merge before their final destiny (intermediate positions), sort of "escape hatches" (Chomsky, 1986; Abels, 2003, 2012). Therefore, accepting successive cyclic movement (a standard approach since Chomsky, 1973), one is forced to accept uniform chains (at least within the orthodox approach, see Chomsky, 2005; Lasnik, Uriagereka & Boeckx, 2005 for examples), that is, all intermediate positions are uniform with respect to a property *p* (e.g., be A / be A'; be a phase-head Spec / be a non-phase-head Spec…). Recently, proposals about the existence (and even necessity) of mixed chains have appeared (e.g., Putnam, 2011; Krivochen & Kosta, 2013) with base on interface conditions, but they have been mostly overlooked. While assuming re-merge from the derivation, mixed chains allow an element to land in intermediate non-uniform sites (e.g., movement from an A- position to an A' position), with different motivations. We will adopt a version of the mixed chains approach, which makes different predictions as to the intermediate derivational steps under a cyclic approach to displacement.

On the other hand, re-merge from the NUM implies that the NUM has access (somehow) to information about how many instances of a lexical item will be needed, implying a massive look-ahead from the NUM to an object that is not yet constructed at the point of NUM selection, by all means, an inconsistency. However, there are alternative ways to save the "re-merge from the NUM" theory. One of them is Stroik's (2009), Stroik & Putnam's (2013) *Survive Principle*, based on the

premise that "if a SO [Syntactic Object] bears any additional features not present on the immediately governing head […], it will survive and remain active in the *Numeration* until all subsequent features have been checked through *Remerge*." (Putnam & Stroik, 2009: 14): that is, *Remerge* is possible as long as a SO has unchecked features (see also Stroik, 2009: 40). While this possibility is conceptually appealing, its implementation requires the additional assumption(s) that:

a) Features exist

b) Features are checked / valued in a specific relation with a head

Needless to say, if a derivation is to build a structure based on the Survive Principle, Merge possibilities will be constrained from the first moment by conditions on feature checking (Boeckx, 2010 follows a similar argumentation, but with different results). In other words, the necessity to create appropriate configurations for feature checking will constrain (re-)Merge possibilities. The scenario thus departs from the simplest one (just free Merge) in more than one significant way. However, we will assume the *Survive Principle* not as parasitic on feature checking operations but as a deep principle of discrete symbols recursive manipulation, and bear it in mind while presenting our *type-token* dynamics.

2.1 Type, Tokens and displacement:

It is commonly assumed that a derivation starts with a Numeration, a subset of the LEX which contains all the elements that will be used in a particular derivation, with diacritics indicating the number of times each element is to be used. For (13), then, we have a NUM like (14)

13) John saw the woman

14) {$John_1$, see_1, T_1, C_1, v_1, the_1, $woman_1$}

Each time an element is introduced in the syntactic workspace, its number is reduced by 1, until it reaches 0: it means that the element is not to be used anymore. We have seen that Survive Minimalism (SM) replaces the numerical diacritic system with feature checking considerations, which are much more pervasive in the Minimalist architecture of the grammar, and not an extra system of elements, as it happens with numerical diacritics. The theory becomes thus much more elegant.

Our proposal shares with SM the concern about how a derivation starts, and how the array of elements to be used is selected. In our terms, a derivation does not start with a NUM, but with a pre-linguistic purely conceptual structure, in the line of Fodor (1975) and, more recently, Jackendoff (2002), Culicover & Jackendoff (2005), Uriagereka (2008), and the sense in which D-Structure is understood in Uriagereka's (2012) CLASH model. That structure is syntactic in a wide sense, as concepts are structured (taking "syntactic" not in the narrow sense of "linguistically structured" but in a strict sense of "structured")[4]. This conceptual structure, shaped by the speaker's intention to convey a certain propositional meaning through linguistic means, is what, in our proposal, drives *Select*, the selection of a subset of LEX, in turn a set of linguistic *types* (this last idea is expressed in Chomsky, 2000 and Uriagereka, 2008). The assumption we make at this respect is the following:

15) Minimal Selection:

[4] Cf. Culicover & Jackendoff (2005: 20 fn. 8): "*Algebraic combinatorial systems are commonly said to ''have a syntax''. In this sense, music has a syntax, computer languages have a syntax, phonology has a syntax, and so does Conceptual Structure. However, within linguistics, ''syntax'' is also used to denote the organization of sentences in terms of categories such as NP, VP, and the like. These categories are not present in any of the above combinatorial systems, so they are not ''syntax'' in this narrower sense. Throughout this book, we use ''syntax'' exclusively in the narrow sense.*" In this paper, and in general within our theory, "syntax" is used in the wider sense, for two main reasons: to begin with, there is no compelling evidence that the "syntactic mechanisms" vary from one system to another (except insofar as the units affect the algorithm, in case that actually happens); and also, an adequately wide formalization of syntactic mechanisms could reveal deep facts about the structure of more than a single system. Admittedly, this requires interdisciplinary co-working and terminology unification, which are not the norm now.

Select the minimal amount of types that can instantiate a conceptual structure CS into a linguistic structure LS losing as few information as possible.

The intuition behind this assumption is clear: we want to instantiate a CS in the most economical way possible[5], *ceteris paribus*. Given the fact that the CS includes not only rough propositional content but also added information (what most linguists would put under the "pragmatics" label: inferences, and other extra-propositional which is, nonetheless, built upon the clues syntactic structure provides the semantic component with), the reference set for each potential derivation is unary: there is one and only one candidate which can express CS in an optimal way. *Select*, then, builds an array of lexical *types* from LEX. At this point already there are important differences with orthodox minimalism:

> *"*(…) *he* [Chomsky] *wants LA to be not a set of lexical types, but rather a set of tokens.* […] *Chomsky wants to identify chains at LF as equivalence relations over the objects in the numeration, but for that he needs lexical tokens, not types."* (Uriagereka, 2008: 16)

Our array is instead a set of *types*, with which we eliminate numerical diacritics in a natural way: a *type* can be instantiated as a *token*, in principle, infinite times, and additional evidence is needed for limiting the instantiation in the syntactic component if the algorithm is purely generative. *Array* is not a primitive concept, or anything like a *representation level*: it is never interpreted, and no conditions on well-formedness apply to it: it is simply a finite set of *tokens*. In this sense, our proposal differs from Martin & Uriagereka's (2011) hypothesis involving *types* and *tokens*. They claim:

> "*There is no natural way of capturing tokens, other than by refying lexical arrays into numerations, objects that exist pretty much in the technical sense of a level of representation*"

Uriagereka's (2008, 2009) *tokenizer* system, which he uses to formalize a notion of recursion different from similar effects that could be obtained via mere *iteration* when we find "an X inside another X" in

[5] In more technical terms, Selection must *reduce entropy*. If the theory of Merge we have developed in past works is correct, the generative algorithm, driven by interface requirements, should also be "counter-entropic".

a linear string (e.g., phonological)[6], differs from ours in a number of technical details, particularly, the formalization of the notion of *type* for syntactic objects (and not only for categories like Case).

Considering this scenario, we should bear in mind that it is a fact of natural language that a type is not unlimitedly instantiated as a token, apart from the theoretical desideratum that operations must be fully explicit and derivations should not contain extra steps (considering DFI and ConsP). Does our theory have a halting algorithm, then? How do we prevent the whole LEX being selected for a derivation, or a derivation which instantiates a single *type* indefinitely (among other computationally and empirically undesirable scenarios)? In two ways: to begin with, recall that the array is selected according to the CS, and following the principle sketched in (15). Therefore, the selection of the full LEX is ruled out from square one. Second, a finite CS cannot be instantiated as an infinite LS, because information would be added, and our principle is to *conserve*, not to *add*, in interactions between modules. Moreover, the CS, the LS and the *array* are mental entities[7], the last two possibly human-specific: objections have already been raised as to the infinite character of the output produced by a finite

[6] The problem whether a linear string […X…X…] is to be considered recursion or iteration (with the concomitant mathematical consequences) and the development of a criterion to establish a straightforward differentiation between them, is currently under research within Radical Minimalism. See Krivochen (forthcoming). For the time being, we refer the reader to the discussion in Lasnik & Uriagereka (2012) and Uriagereka (2012).

[7] We briefly address an interesting issue raised by Thomas Stroik here: how come, if language is indeed a physical system, as we claim, that there are types and tokens, or blind Merge, something that is not found in other physical systems or processes (say, for example, the formation of a complex molecule)? Two aspects of the question are to be considered: on the one hand, our main thesis is that all physical systems are identical "*at a principled level of abstraction*", not at a substantive level. By "principled level", as we have argued in previous works, we mean architectural issues, particularly as it comes to complexity. Therefore, we argue that any kind of "complexity" can be studied formally as an interpretative epiphenomenon of the interaction between simpler units, until getting to the unit that integrates another but is itself not composed by any other. On the other hand, and relatedly, the formation of a complex molecule is indeed limited by the characteristics of the units that are manipulated, that is, chemistry is not crash-rife, and there are compounds that are either unstable or directly do not appear in natural conditions. That is not due to a property of the union (e.g., covalent, ionic, metallic), but of the elements involved. Contrarily to endoskeletal models (GB, Minimalism, HPSG, LFG), it is not the properties of lexical entries that determine the shape of the syntactic configuration, but semantic interface requirements, precisely because we do not have lexical entries as such, but roots and procedural elements, if at all (see Krivochen, in preparation, for discussion).

substratum (e.g., Katz & Postal, 1991). The conservation principle we mentioned above has been stated in previous works as follows:

16) Conservation Principle

Dimensions cannot be eliminated, but they must be instantiated in such a way that they can be read by the relevant level so that the information they convey is preserved.

The property of displacement reduces to *External Merge a token from the type-Array* if we consider that a licensing predicate can trigger the E-Merge of an argument token in order to fulfill Interface Conditions, the Conservation Principle (ConsP) and, as far as step-by-step derivational justification (i.e., why should the generative algorithm apply once again?) is concerned, Dynamic Full Interpretation (DFI):

17) Dynamic (Full) Interpretation

Any derivational step is justified only insofar as it increases the information and/or it generates an interpretable object for an Interface Level IL.

The number of tokens required is determined by interface conditions, so that the *minimal number* of tokens leading to a convergent object is used, provided that the notion of "convergent object" does not arise from look ahead (the syntactic component looking at the legibility conditions of the interface systems, which would lead us to a "bad" crash-proof system, in which a partly interpretative system makes interface requirements superfluous: a *constructivist system* of the kind defended by Lasnik, Uriagereka & Boeckx, 2005 Chapter 2; or Frampton & Gutmann, 2002), but rather from the interfaces "peering into" the syntax (something similar to the proposal by Boeckx, 2007, although the consequences he draws are completely different from ours) and *Analyzing* after every application of

Merge whether a syntactic object is ready to be transferred, resulting in the following derivational dynamics:

18) Concatenate (α, β) = {α, β}

Analyze$_{IL}$ {α, β} [is {α, β} fully interpretable by the Interface Level IL?]

(Transfer {α, β]} to IL if Analyze$_{IL}$ results in convergence at IL)

The idea of "invasive interfaces", in our framework, is a natural result of separating interpretative and generative systems. If generation is restricted to a single operation *concatenate* (Cf. Hornstein & Pietroski, 2009; Boeckx, 2009, who include *Label* as concomitant to the structure generating algorithm), which is the optimal scenario (since there are no *a priori* Agreement restrictions, Cf. Pesetsky & Torrego 2004) and it occurs in an *n*-dimensional workspace (another null hypothesis: restricting dimensions to 2, as in traditional Kynean tree diagrams, is stipulative, derived from considerations of "unambiguous paths" and linearization issues, see Kayne, 1984, 1994), it is only natural that the operation cannot either *read* or *evaluate* what it has built. On the other hand, it is also only natural that the EVAL function (in OT terms; *Analyze* in ours) is not separated from the interfaces but in itself be *the set of Bare Output Conditions* (or legibility conditions) each cognitive interface has. Then, if we depart from considerations of computational simplicity like "maintain as few structure at once in W as possible", that is, "transfer as soon as you can" (bearing an obvious resemblance to Pesetsky's 1995 *Earliness Principle*). "As soon as you can" is determined not by internal syntactic conditions (as they do not exist in our proposal), but by *Analyze*, which is, as we have seen in (18), an integral part of the derivational dynamics. In this way, the generative workspace is wiped clean several times along a derivation (without this being pre-determined by the presence of a certain element –say, a phase head-) thus liberating working memory without the concomitant problems of defining, for example, endocentric transfer domains or fixed "contextually defined parts

of a derivation" (Grohmann, 2003: 75), since part of that "context information" involves Agr issues that in turn rely on feature valuation processes.

3. Implementing the machinery

The last section was devoted to the description and theoretical justification of the advantages of adopting a *token* approach to displacement phenomena, instead of a Copy+Merge theory. In this section, and in the rest of the paper, we will put this system in practice. The focus of our inquiry will be set on *parasitic gaps* in English and Spanish. The aim of this section is to provide the reader with tools to apply the framework to phenomena of his interest.

Let us see an example of how we would derive with our system. Consider the (simplified) array in (19):

19) {Who, come, T, C}

And the derivational point (20):

20) [$T_{PastPerf}$ [V come [Who]]]

There is an element, T, which generates drastic interface effects at two levels:

a) Via its own Time specifications
b) Via thematization of an element merged within its projection

The Time specifications do not require, in this case, any further operation (since finite Tense is absolute when in a matrix clause), but there is the possibility of thematizing the subject of the Unaccusative verb [come]. How do we do it? By merging a token of the relevant element extending the structure, which would result in a phrase marker along the lines of (21):

21) [Who [$T_{PastPerf}$ [V come [Who]]]]

What is the difference between a token theory and a copy theory? To begin with, a copy theory (both Chomsky / Nunes', and Stroik's) requires an additional Copy operation, something ontologically different from Merge: Merge applies blindly to *n* objects and forms sets, Copy applies to *one* object and its product is not describable in set-theoretical terms until a chain is formed, presumably at the interfaces: up to that point, there being no evidence that the syntactic object SO and its copy SO' are linked, there is no dependency between SO and SO'. Secondly, the apparent advantages of copy theory (e.g., elimination of indexes, as Chomsky, 2000: 114, 145 fn.62 claims) are actually not so when one considers the claim that operations are feature-driven (directly or indirectly, a further and unclear distinction, Chomsky, 2000: 107-8). Let us assume that [who] has an unvalued feature/s (say, Case) in its merge position. If operations are feature-driven, that means that movement will be aimed at valuating that feature so that the derivation doesn't crash at LF. However, if the system is to link copies of an element, there is a problem: copies are not identical. The feature matrix of the lower copy contains an instance of a feature which the higher copy does not. The problem with this approach can be exemplified using Nunes' (2004) mechanism for chain interpretation:

22)

- Copy
- Merge
- Form Chain
- Chain Reduction

The first element to take into account is that the four steps are independent operations. This means that each must have an independent justification, apart from their relevance to Move. Merge is arguably justifiable from "conceptual necessity" (see above), but it is difficult how Copy, Form Chain and Chain Reduction can have independent interface justifications. Moreover, it is not clear where each step occurs, that is, at which derivational point or if at the interfaces. Copy and Merge occur in the

syntactic workspace, apparently, but there is no way Form Chain can occur in the syntax, since it involves establishing a dependency between constituents, which is something that can only occur at an interpretative system, not in a workspace where a purely generative algorithm applies. This operation synthesizes, however, Chomsky's and Rizzi's approaches, as Chain Formation requires *identity, c-command, feature relation* and respect of *Minimality effects* (Rizzi, 2004). Identity, however, is not defined, and the same criticisms that we have made above apply here: identity is not possible in a system that admits feature checking / valuation relations, since the feature matrix of an element varies as the derivation unfolds (in fact, for Chomskyan oriented theories, the derivation unfolds *precisely* because feature matrices have to get rid of uninterpretable features via Agree) and copies are (Internally) merged in places in which they can value and erase uninterpretable features. The relevant interface cannot establish a dependency between two objects, say, α and β, such that:

23) $\alpha = \{i\text{-}F_1, i\text{-}F_2, u\text{-}F_3, u\text{-}F_4\}$

$\beta = \{i\text{-}F_1, i\text{-}F_2, \sout{u\text{-}F_3}, \sout{u\text{-}F_4}\}$

Let us assume the features involved are the following:

24) F_1: D

F_2: ϕ

F_3: Case

F_4: Wh-

In the first-Merge position (say, Compl-V), the element has the feature matrix α, whereas after checking Case with *v** or T and movement to Spec-CP, the matrix is β. We see that the identity criterion is not met: feature matrices are different. C-command depends on a 2-D phrase structure model, against which we have already argued (Krivochen, 2012b, in preparation). The very feature

valuation system provides arguments against itself, and definitely against establishing dependencies in a non-stipulative way. We have already proposed a solution to the problem of the feature system by eliminating the concepts of features and values (Krivochen, 2011), as they are only supported by intra-theoretical stipulations (the "imperfections" Chomsky 1998 posits as a way to circularly justify the need for feature-valuation operations), and the identity problem by positing a token-merge theory of Movement. In the next section, we will confront our proposal with previous ones, particularly that of Nunes (2001, 2004) involving *sidewwards movement* (or inter-tree dependencies), which has been adopted by Kayne (2005), Boeckx & Hornstein (2004), Wiland (2009), and Hornstein (2001, 2009), among others.

3.1 *(Parasitic) Gaps revisited:*

Transformational theories of syntax, faced with apparently non-uniform displacement phenomena, came up with a distinction that has become almost commonsensical in current syntactic thinking: it is assumed in the literature (e.g., Chomsky, 1982, 1986; Engdahl, 1983; Kiss, 1985; Nissenbaum, 1998, 2000; Nunes, 2004 among many others, see Culicover, 2001 for an overview) that there are two types of gaps (i.e., empty positions left by constituent displacement):

a) True Gaps (TG): originated by (A-A') movement of elements (either arguments or adjuncts) within a tree, giving rise to a successive cyclic chain respecting traditional constraints on displacement and dependency establishment (Subjacency, Relativized Minimality, Condition on Extraction Domains, Minimal Link Condition, etc.). TG are interpreted by the presence of an operator binding the variable within local domains, and do not need to be licensed by any other chain.

b) Parasitic Gaps (PG): licensed by a TG in a syntactic object, within which the PG and the TG are coindexed. Crucially, as we will see below, PG are traditionally claimed to appear in subordinate adjunct clauses, from which extraction is banned. We will expand on PGs, as they are the focus of the present section.

Some commonly assumed properties of PGs are the following:

- PG are licensed in S-Structure (i.e., after the application of a transformational rule)
- Antecedent in an A'-position in the matrix clause: PG cannot be licensed by A-chains
- PG are always DPs
- PG cannot be licensed by "non-referential NPs" (Nunes, 2004)
- TG *cannot* c-command PG, as PGs appear in adjuncts (Chomsky, 1982; Taraldsen, 1981)
- The adjunct must be non-finite (most commonly, infinitival)
- PG chain and TG chain constitute two different chains, in order not to violate the CED (Chomsky, 1982, 1986; Nissenbaum, 2000; see Nunes, 2004 for a different view on the matter).
- PGs are licensed by leftwards movement (Postal, 1994)

In this section, we will see some instances of gaps that cannot be classified as either TG or PG according to traditional criteria, which should be revisited in order to gain descriptive and explanatory adequacy. This, we will argue, is achieved via the *External Token Merge theory of displacement* we have explained above. Let us review a classic example of PG in English:

25) Which paper$_i$ did you file *TG*$_i$ [without reading *PG*$_i$]?

We see that all the characteristics are present here. There is A'-movement to Spec-CP and a PG licensed by this A'-dependency in an adjunct (as PGs are in adjuncts, regardless their linear position, they never c-command the TG). In this case, Nunes' *Sideward Movement*, which assumes only one chain, but with inter-tree dependencies, or the more traditional explanations involving an empty $Op_{[Wh]}$ in Spec-CP within the non-finite adjunct, and a *trace* in the PG position apply and are descriptively equivalent: the difference relies on the number of intermediate landing sites. While Chomsky's 1982 solution required three empty categories for an example like (25), Nunes' 2004 and related proposals require one *per* phase head, depending on the author's conception about phasehood:

26) [Which paper]$_i$ did you file t_i [Op_i without PRO reading t_i] (Chomsky's proposal)

27) [CP [Which paper] [did [TP you [*v*P ~~which paper~~ [*v*P [*v*P file [VP ~~file~~ ~~which paper~~]] [PP without [CP [TP PRO [*v*P reading ~~which paper~~]]]]]]] (Nunes' proposal)

The operator in (26), in due time –at LF, perhaps-, is (somehow) coindexed with the Wh-element in the matrix clause, which allows the interpretative component to establish a dependency and satisfy Full Interpretation, as no element is left uninterpreted.

After this brief summary, let us analyze the following structures in Spanish and English:

28) Quedó media botella de cerveza [sin tomar *t*] (TG/PG?)

There-is half a bottle of beer without to-drink t

29) a. Tengo todavía muchas fotocopias [por leer *t*] (TG/PG?)

Have$_{1Sg}$ yet many photocopies left-to-read t

b. I have many photocopies left [to read *t*]

We see that in these examples there are so-called "extractions" from an adjunct PP or non-finite clause (which clearly violates the CED and other orthodox takes on locality via *impenetrability*) and there is no other gap (either A- or A') that can be said to legitimize the dependency, in traditional Minimalist terms. Moreover, there is no chain at all to license the gap, in case it is taken to be parasitic. Notice that verb typology cannot be appealed to in order to explain the phenomenon, since we have an unaccusative existential construction in (26) and a transitive, accusative construction in (27). The distinction TG/PG, then, must be deeply revisited.

The question we must ask ourselves now is: "does TG/PG distinction make any sense"? Let us revise what we have said so far:

A "true" gap is where we expect "regular extraction", whereas a PG is a structural place where extraction is usually not possible without the presence of a licensing TG. Categorially, nevertheless, they both behave like Wh-traces within their own projections (and thus the proposal of a null *Op* in the non-finite clause to bind the PG in Chomsky's account). This apparent unification (due to Michael Putnam, p.c.) has a disadvantage, however: one still has traces and properties assigned to them without counting, of course, the mere concept of "extraction", i.e., the vision of *movement* as *displacement* against which we have argued. The provisional conclusion is that, rather than conditions over extraction (which have been mostly subsumed to the availability of suitable *probes* in a feature-valuation driven framework), we should look for conditions over the establishments of dependencies between constituents *at the interfaces*, as we have proposed to do via *Token Unification*, and leave the GEN function free and unbounded. The possibility of applying a *type-token* theory is, we believe, essential to explain the semantics of the so-called PG structures, since the TG-trace and the PG have crucially *the same referent* (formalized via diacritic indexes), which is a problem for traditional approaches if one follows the claim that reference is established via c-command of *Op.* over *variable* (e.g., proper name-common DP).

Our system, thus, *Merges-α* freely and *dependencies* are established at the interfaces, where interpretation procedures apply. In our case, the establishment of a dependency has to do with the relation the interpretative component C-I can establish between the set of coordinates that define the position of α and β in generative workspaces, what we will formalize via *token unification*. As far as $C_{(HL)}$ is concerned, there are objects sharing format that are merged following ConsP and DFI, both interface requirements at global and local levels respectively. In this sense, both PG and TG are derived possibly in parallel by a simple mechanism, which can be made explicit as follows:

30)

- Assemble a number *n* of *token-α* of the relevant constituent (either atomic or complex) in W_X, being *n* the minimal amount of tokens required to generate a specific semantic effect.
- Merge *token-α* in the matrix structure, assembled in W_Y.
- C-I Interface establishes a dependency between α and β *iff definitions 24-27* obtain.

With this framework, we can better explain multiple problems, like the so-called "resumptive pronouns" and *weak island violation*. Let us analyze the following examples (taken from Alexopoulou & Keller, 2003: 3):

31) **No island violation**

a. Who does Jane think that Mary claims that we will fire *t*/him?

32) **Weak island violation**

a. Who does Mary wonder whether we will fire *t*/him?

Take, for example, (32), which is problematic for orthodox theories of locality as either impenetrability or intervenience: the impenetrability condition *par excellance* is the so-called Phase Impenetrability Condition (Chomsky, 1999, 2000), which renders complements of v_0 and C_0 invisible for the purposes of further computations. Notice that, if [whether] is in Spec-CP (see, for instance, Emonds, 2007: 378 fn. 33), an intervenience approach to movement would block any dependency between [who] and either a trace or a resumptive pronoun, since there is a potential governor in between. If, on the other hand, [whether] is a C head, then it must be a phase head, thus triggering Transfer of its complement, and/or making it opaque to operations at the next phase head (it all depends on the version of the PIC one considers). In any case, impenetrability or interveniency, there should be no way of establishing a dependency between the resumptive pronoun and the Wh-word, which would leave an uninterpreted element in the final representation. Clearly, an alternative solution is to be sought for, particularly taking into account that the version without the resumptive pronoun is decidedly less acceptable (if acceptable at all) for native speakers.

In our terms, all traces are eliminated in favor of tokens to be unified at the C-I interface following the procedure specified above, and resumptive pronouns are taken to be Spell-Out *tokens* of the same DP *type*, *á la* Grohmann's (2003) *Copy Spell Out*, which allows a modification of the phonological exponent of a terminal node if there is intra-domain movement (a condition we will eliminate, as the reader will notice):

33) Who does Mary wonder whether we will fire ~~who~~ ⇨him

Now, why should we bother in spelling the pronoun out? Because, for the purposes of parsing, *reference* chains require what we will call, borrowing the term from information theory, *signal reinforcement*: the origin of the "signal" is the first occurrence of a D-*token* in the linear order [who], and the *intensity* of that signal depends on the informational "weight" of the aforementioned [DP]

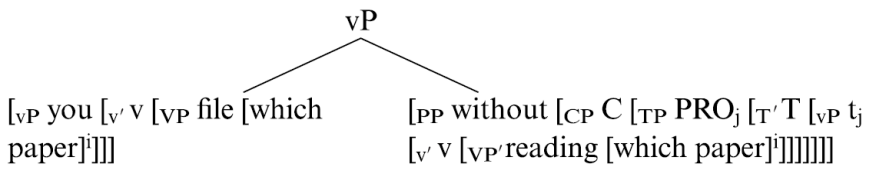
token. In relevance-theoretic terms, the defining factors are *prominence*, *internal complexity* and *structure* of the ostensive stimulus (Wilson & Sperber, 2003). A [DP [CP]] structure (e.g., a D with an embedded relative clause) is "heavier" (even in traditional terms) than a simple [DP D [N]] structure, and heavier DPs are more difficultly forgotten by the parser. Thus, the informationally heavier the DP (and the more complex syntactically), the longer it will stay in the active working memory and the less the need for an immediate *signal reinforcer*, by which we mean simply *another Spelled-Out {D}-token*. A point in favor of our analysis is the fact that native speakers explicitly prefer the version with the resumptive pronoun, so that they can retrieve the reference of the variable in Compl-V.

The advantages of our analysis over, for example, that of Nunes (2004), are quite straightforward: let us take an example like (34):

34) Which paper did you file without reading?

Nunes' derivation of that parasitic gap supports our multiple workspace theory (already proposed for so-called "coordinate gaps" by Williams, 1978; see Engdahl, 1987), since the complex object in (35) could not have been formed by *monotonic Merge* (Uriagereka, 1999/2002; Krivochen, 2011):

35)



At this respect, Nunes says:

"*As the derivation proceeds, other lexical items are pulled out from* [a Numeration] *N' in (24a) and merge with K and M* [terms formed by Merge from N and N'] *in (25), yielding the objects P and Q in (26).*

(26) a. P = [$_{PP}$ without [$_{CP}$ C [$_{TP}$ PRO$_j$ [$_{T'}$ T [$_{vP}$ t$_j$ [$_{v'}$ v [$_{VP}$ reading [which paper]$_i$]]]]]]]

b. Q = [$_{vP}$ you [$_{v'}$ v [$_{VP}$ file [which paper]$_i$]]]

(27) [our (35)] *represents the next derivational step, where P adjoins to Q* […]. *In (27), no chain formation between the nondistinct copies of which paper can take place, since they are not in a c-command relation (...)*" (2004: 99)

We can see that his analysis needs to resort to parallel derivational workspaces, just as our analysis requires, but with the added complication of a trigger for *sidewards movement*, establishing either inter-tree dependencies or determining the derivational point in which a structure is non-monotonically merged. Moreover, the approach is grounded on c-command requirements, assuming an LCA framework and the mechanism of *chain reduction* Nunes proposes (2004: 27). So, from the four steps Nunes requires for *sidewards movement* to be licensed in the syntax, we only leave one in the syntactic working area (free concatenation) and the "chain formation mechanism" is replaced with periodical *token unification*, at the interfaces. Nunes' C+T theory (in turn, heavily based on Chomsky, 2000 and Uriagereka, 1999/2002) can be summarized as follows:

36) Nunes' (2004: 89) Copy + Merge theory of movement:

a. Copy
b. Merge
c. Form Chain
d. Chain Reduction

Our *token*-Merge theory of displacement allows us to dispense with "Copy", since all we have to resort to is External Merge, and there is, crucially, no difference between TG and PG as to their derivation. Once a syntactic object is transferred, the relation between coordinates of terms results in a dependency, as we have defined them. A derivation of a PG, in our model, would replace Nunes' (26)

with (37), more in line with Hale & Keyser's (2002) and Mateu Fontanals' (2002, 2005), among others, proposals on lexical conflation:

37)

a) [*v*P you [*cause* [VP [GO] [PP which paper [[TO] file]]]]] ⊂ W_1

b) [P+Neg [*v*P you [*cause* [VP read [DP which paper]]]]] ⊂ W_2

A DP is assembled via monotonic Merge in W_1, and another token of that type enters the derivation in W_2, when a licensing node (e.g., P or V) requires so to generate a drastic interface effect: notice that a P establishes a relation between a Figure and a Ground, and if there are no entities to relate, the P becomes a superfluous element in the representation. Technically speaking, we have two distinct objects in a multidimensional generative workspace, the dependency is only established *at the semantic interface* (accessed cyclically, following the derivational dynamics made explicit in (18) above), and the choice of materialization copy depends on Spell-Out patterns in a language L, following a minimal effort desideratum:

38) *Spell Out the minimal amount of tokens of each type needed to satisfy interface requirements*.

This proposal clashes again with Nunes' (2004: 44) Chain Reduction Principle:

39) *Delete the* ***minimal number*** *of constituents in a non-trivial chain CH that suffices for CH to be mapped in a linear order in accordance with LCA* (our emphasis)

While we attempt to minimize Spell-Out, focusing on the effects each token has at the semantic interface, in a monostratal theory; Nunes' account, focused on the realization of chains at PF, depends on the LCA, in turn parasitic on the notion of asymmetric c-command (only formulable in 2-D tree-like structures). If our system explicitly impoverishes syntax to a single *n*-ary concatenation function,

only limited by semantic requirements (ConsP and DFI), such constraints are not even formulable in our theory. The interpretation of both instances of [which paper] as a single entity is also a natural consequence of adopting a *type-token* dynamics: if *n* constructions (of arbitrary complexity) are instantiations of the same type(s), then they denote the same coordinates in the conceptual space. In short, they share *denotatum*[8]. In any case, both syntactic objects in (36) can then be instantiated as a single unit in W_3 and free concatenation applies: the procedure is analogous to that which generates complex predicates of the type of (40):

40) A *hard-to-do* work

In the case of (40), the whole structure [to do *e* is hard], generated in a separate workspace, is instantiated *as a single root* in the workspace containing [DP A work], and via *token unification*, *e* and [work] are identified as tokens of the same type. The same reasoning can be applied to the derivation of parasitic gaps, which require multiple parallel workspaces.

Our framework allows us to tackle examples like (41), to the best of our knowledge, considered "peripheral" in orthodox generative grammar:

41) This is a problem which John is tough to talk to about (From Green, 2011: 38)

HPSG treats such non-coreferential gaps by means of lexically specified dependency features, as did GPSG. Having argued against the implementation of all-powerful features, we must find another way out for multiple gap sentences like (41). Let us first identify the relevant gaps:

42) This is a problem$_j$ which$_j$ John$_i$ is tough to talk to e_i about e_j

Let us now chunk the representation into separate assembled objects, materializing all tokens:

[8] This system allows us to define anaphors, for example, as spelled-out tokens of the same type within very local boundaries (see Krivochen, in preparation).

43) a. This is a problem

b. John is tough to talk to John

c. about a problem

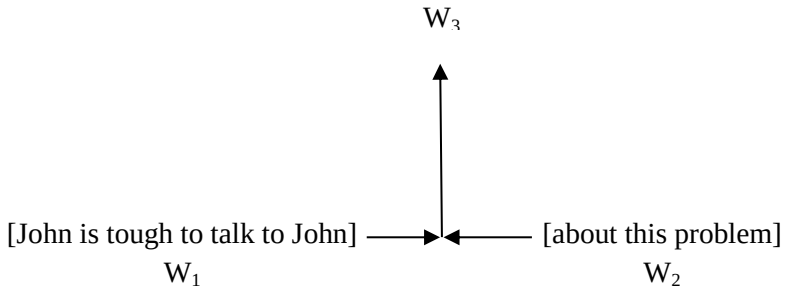
The derivation of the object in (b) posits some problems, already addressed in Krivochen & Kosta (2013, Chapter 4), to which we refer the reader. That kind of complex predication, for the purposes of this paper, will be analyzed as adjectival (without decomposing A further, even though A = P + N, following Hale & Keyser, 2002 and much related work). Notice:

44) John is [tough / nice / smart]

45) *John is [to talk to]

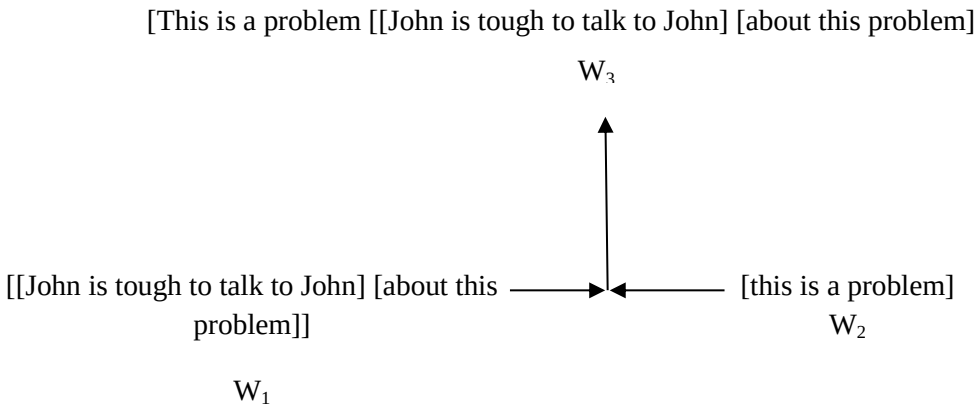
Going back to the derivation of (42), the on-line processes in parallel workspaces generate the objects in (46):

46) [[John is tough to talk to John] [about this problem]]



So, both objects are unified in a W_3 (which, for the purposes of the present discussion, could be identical to either W_1 or W_2). The complex object derived in (46) is in turn merged with the object in (43 a):

47)



Now we have assembled the full object in (41) in the syntactic workspace, and C-I can interpret the object with no inconveniencies. However, there are still Spell-Out issues to be explained: how do we get the final PF? Why can [a problem] receive a double Spell-Out (as Ø or as [which]) in the relative clause and there is no possibility to materialize [John]? Largely, this is a matter of semantics, since our "lazy Spell-Out" desideratum, informally stated as:

> *Spell-Out as few elements as needed for convergence, unless there is a powerful inter-face reason to Spell-Out elements that are not strictly necessary for plain convergence* (Krivochen & Kosta, 2013: 178)

What we understand by "plain convergence" is simply "full interpretation with-out any extra interface effect like implicatures, presuppositions and the like". Extra-positive cognitive effects, which require extra processing work, balance the Relevance ratio, but this does not mean any positive cognitive effect is actually relevant: far-fetched propositions, if required to create a context of intelligibility for an LF, drastically reduce Relevance and occupy working memory. This proposal contrasts with Nunes' Chain Reduction insofar as we attempt to minimize the number of copies based on what the subject interpreting our message might need to construct a mental representation of the CS we wanted to convey linguistically.

These considerations allow us to dispense with the token of [John] in Compl-P: if we spell it out, not only there is no additional positive cognitive effect to be extracted (Sperber & Wilson, 1986; Wilson & Sperber, 2003), but processing cost is increased with no benefit in sight. The problem, if there is actually one, resides on [a problem]. In the next section we will see how the mechanism of *token unification* at LF allows the system to interpret […X…X…] as two tokens of a single type and not two distinct syntactic objects altogether. Spell-Out of both [a problem] tokens is optional, but what is sure not is that both tokens cannot receive the same phonological exponent. That is, in case the referential chain is to be reinforced by the materialization of its two members, the phonological exponent *has to differ*. This does not guarantee convergence, surely, but is a good starting point. Consider:

48) John cut John

The only natural (i.e., non-forced) interpretation is disjoint reference: both instances of John are instantiations of *different types*. If, however, both instances are *tokens of the same type*, Spell-Out assigns a different phonological exponent to the "lower copy", thus getting "John cut himself"[9]. A Natural Language NL has a limited inventory of form to materialize different kinds of constituents, and choice largely (if not entirely) depends on syntactic context: given two (or more) referential variables (i.e., the two tokens of [a problem]), the system has two choices to make:

a) Is it necessary to Spell-Out more than one?

[9] In Krivochen (in preparation) we suggest that LEX is composed of Δ type-variables, whose Spell-Out depends exclusively on the syntactic context. The variables enter the derivation (instantiated as *token-variables*) in a ψ-state as far as its phonetic form and interpretation are concerned: an anaphoric or a pronominal form will surface depending on the presence of another Δ variable in a local domain. The principle we formulate in that work is the following:

∀(Δ) | Δ ∈ D [D an interface-defined domain], Spell-Out(Δ) =

- Pronoun *iff* ∄(Δ') | Δ' ∈ D & Δ ∈ Δ'
- Anaphor *iff* ∃(Δ') | Δ' ∈ D & Δ ∈ Δ'
- RD *iff* ∄(Δ') | Δ' ∈ Σ [a derivation] & Δ ∈ Δ'

b) If it is, which phonological exponent is to be assigned to each?

In this case, the Spell-Out of both tokens of [a problem] is optional, and 10 native speakers consulted have suggested that the contact relative clause version (i.e., without materializing the second token of [a problem] in a relative Wh-word form) sounds "more natural". Given the fact that no extra positive cognitive effects are obtained from the extra materialized token, our Anti-Spell-Out generalization would lead the system to answer "no" to the first question (metaphorically speaking, of course).

A note is in order: the reader must have noticed that we argue for no transformational rule linking "hidden" and "patent" levels of syntax, nor are derivational steps syntactically dependent on each other. The system we argue in favor of, then, is not only *monostratal* (since multiple parallel workspaces is not the same as multiple representations which derive from each other via the application of an algorithm, as in GB; or several levels linked by mapping rules, as in Williams' 2003 *representation theory*) but also *non-transformational*: the *type-token* distinction allows us to maintain descriptive adequacy while eliminating the theoretical machinery necessary to account for *empty category* interpretation (see Krivochen & Kosta, 2013 for details and discussion).

3.1 Unifying Tokens:

In the previous sections we have presented the possibility that, instead of *copies*, we have a *type-array* from which *tokens* of each element are introduced in the W to be manipulated by the generative algorithm *concatenate*. There was a question left unaddressed:

- How does a type-token system deal with interpretation at LF and Spell-Out?

To begin with, we will make a fundamental assumption in our framework explicit: contrarily to both Endo-Skeletal (NSM) models (according to which clause structure is a projection of lexical requirements, as in the lexicalist GB tradition) and Exo-Skeletal (XSM) models (according to which

clause structure pre-exists a particular derivation, as a universal hierarchy of functional projections to be filled with lexical items, our on-line generator departs from the claim that, for a structural position *P* to be licensed in a derivation *at a certain point*, there must be an (procedural) element *e* licensing that position, such that positions are licensed in real time as elements are introduced in the derivation. Formally,

49) $\forall(P), \exists(e) \mid e(P)$

However, the fact that a position is licensed *does not mean it has to be filled*. For example, in a sentence like (50)

50) A mí me parece que Juan miente (Sp.)
'To me CL_{DAT} seems that John lies'
It seems to me that John lies

the dative clitic licenses the PP [a mí], which can be proven via a simple elision test:

51) a. Me parece que Juan miente
'CL_{DAT} seems that John lies'
b. *A mí parece que Juan miente
'To me seems that John lies'

As we have argued elsewhere, it is the clitic that licenses the presence of a PP, not the other way around (Cf. Gutiérrez Rexach, 2000: 316). However, it is possible that the licensing element is present, but the licensed element remains covert, if present at all. Considerations of derivational and representational economy lead us to think that, in this case, even though the Topic position for the PP is *licensed* at LF (that is, it would be interpreted if transferred), it is not projected in (51 a). We stipulate no condition over the generator, we only propose DFI as an on-line, step-by-step *evaluator*

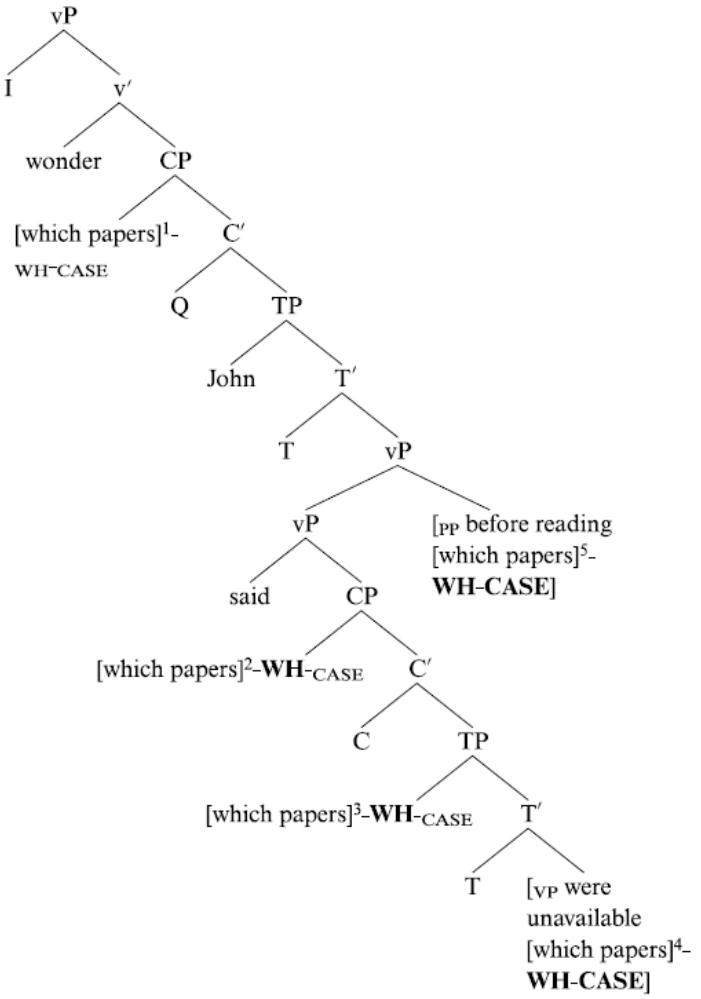
that is interface-driven, with our focus set on LF. The answer to the first question, then, is that types are selected to be instantiated as tokens *if and only if* there is a procedural element (P, T, *v*, Top, Foc, Asp, Mod, D) that licenses the presence of the relevant token. This is a necessary and sufficient condition, but, as we saw, the fact that a position is licensed does not mean it is actually projected. We depart, in this point, from traditional X-bar theory, and gradually, from both Spec- and Adjunct proposals regarding the status of adverbs, both parasitic on the X-bar scheme in one way or another.

The second question is a bit more complicated, since it requires a fully explicit theory of how interfaces select the token to be materialized, and prevent meaning atomizing, which would be a direct consequence of a naïve application of the type-token theory. That is, how can we merge a token of a type in different positions, having a blind syntax, and then pretend to have a full interpretation of the relevant token, unifying the interface effects associated with each position? An example may clarify the problem just posited. Let us consider the sentence in (52):

52) I wonder which papers John said were unavailable before reading (Nunes, 2004: 113)

Nunes' analysis goes along the lines in (53), following his proposal of Sidewards Movement (which has become mainstream, replacing Chomsky's 1982, 1986 proposal of an empty *Op* in the parasitic gap, co-indexed with a trace within the adjunct clause):

53)



(Nunes' (64), 2004: 125)

Notice the multiple occurrences of the QP [which papers]: let us analyze whereas all are interface-motivated by semantic requirements of a given head or their occurrence obeys intra-theoretical stipulations. Nunes assumes the Chomskyan theory of phases, according to which the derivation is "chunked" into smaller bits, defined by the presence of φ-feature bearing heads C and *v**. The complements of those heads, under standard assumptions, are impenetrable to operations from outer probes, any outer probe in the stronger versions (Chomsky, 2000) and the next phase head under

weaker assumptions (Chomsky, 1999). The relevant condition is the so-called "Phase Impenetrability Condition":

> *The domain of H is not accessible to operations at ZP* [the next strong phase head], *but only H and its edge* [Specs and Adjuncts]. (Chomsky, 1999: 29)

Nunes' occurrences of the DP [which papers] is purely determined by phase-theoretical requirements[10]: notice that they appear in Spec-positions of *v*Ps and CPs in order to allow further cyclic movement, except the intermediate step in Spec-TP, which remains unexplained: why is the Case feature of the DP [which papers] left unvalued? If TP is dominated by CP, according to the theory outlined in Chomsky (2005), TP should have inherited the relevant Tense and φ-features to value the Case dimension in the DP. However, the constituent keeps moving, perhaps because of some phasal requirement of a higher head. The problem with this kind of reasoning is that it is incompatible with an on-line derivational system, yielding a strong representational theory instead, in which the computational component can have at a derivational point T access to information about non-yet formed objects. It is also curious how copies 1, 2, and 3 have their Case feature valued, but 5 does not, thus containing an uninterpretable feature which would make the derivation crash.

The chain is also uniform (following the dictum of Chomsky & Lasnik, 1993), provided that the relevant property is "phasehood" of the licensing head. This is admittedly not connected with interface requirements, as Chomsky (2005, 2007) claims: the most important criterion in determining phases is not related to interface properties, but to the Case/agreement systems. In particular, reliance on the latter allegedly make phases follow "in a clear and self-contained fashion" (Gallego, 2010: 54), since

[10] …and stipulations. To this day, it is not clear why unaccusative and passive VPs are not regarded as "strong phase heads", or the criteria that has been used to determine the identity of strong phase heads. If the presence of inherent φ-features or uninterpretable features of whatever nature (as Gallego, 2010 suggests) is the relevant factor, it is even less clear why the aforementioned constructions do not count as phase heads, and why should transitive *v*Ps be endowed with uninterpretable features if not for theory-internal requirements of Case valuation on c-commanded DPs.

deletion of uninterpretable features force phase boundaries to emerge, a view systematized in Gallego's (2010: 51) *Phase condition*. Needless to say, this proposal strongly contrast with ours, which defines domains in terms of fully interpretable units for either SM or CI, and explicitly claiming that features play no role in the Narrow Syntax (or anywhere else). Since both interfaces can access the generative workspace, it is only logical that we must distinguish two kinds of phases, or local domains: PF local domains (i.e., *analyze* looking for tone units, for example) and LF local domains (i.e., *analyze* looking for locative, eventive-causative, and *modus*-oriented domains, as well as fully-fledged sortal entities capable of undergoing a *referent assignment* process at LF (see Wilson & Sperber, 2003 for details). In Chomsky's proposal (and related views), there are at least as many instances of a SO as phase heads (both for impenetrability *and* feature valuation reasons); in ours, there are, at most, as only many tokens as licensing nodes.

The consequences of this formulation for the theory of successive cyclicity are clear: there will be only as many tokens as minimally required to generate the interface effects determined by CS. Periodicity in punctuated paths (see Abels, 2003, 2012 for discussion) is subsumed to interface requirements, such that not only is it stipulative but also undesirable: as we have argued in Krivochen & Kosta (2013, section 4.4.2), uniform chains in the sense specified above are trivial, insofar as the informational load is not increased (within the limits of CS) through the course of the derivation. Following DFI, Merge (and *remerge*, being a particular instance of Merge) applies if and only if the informational load is increased. If all positions are licensed by heads which are uniform with respect to a certain property (stipulatively determined), the fact that a SO of arbitrary complexity is (re)Merged in a certain position will contribute nothing to generate interface effects. In more explicit terms, this is what we have called the Condition on Mixed Dependencies (Krivochen & Kosta, 2013: 126):

54) Condition on Mixed Dependencies (CMD)

A dependency between tokens must involve different properties in each structural location in order to legitimize the operation of Concatenation in interface terms. A uniform dependency is trivial at the interface levels.

In other words, a dependency between syntactic of arbitrary complexity objects is legitimated at the interface levels if and only if it follows from a derivation built respecting DFI. The question now is: which positions are relevant at these effects? Thematic theory considerations would lead us to say that *three* is the minimum amount of required tokens: one should be merged within the matrix VP to be theta-theoretically interpreted as a *theme* (or, in locative terms, a *figure*), identified with a 1. Being accessible by finite T, the DP1 in question could also receive a Nominative Case interpretation at C-I, following the configurational proposal of Krivochen (2012) and Krivochen & Luder (2012). Another token is minimally necessary in the outer position licensed by a head in the left periphery, say simply C, to generate the Focus/Comment interpretation. This token, we will identify with number 2. Finally, within the adjoined clause, a third token in the VP domain is necessary for the final interpretation of the *collapsed token* to include both "Compl-were" and "Compl-reading" information, that is, the same papers which were unavailable were those John did not read by the time of utterance. The structure we propose is, then:

55) I wonder [which papers2 John said were unavailable which papers1 [before reading which papers3]]

There being no EPP feature to satisfy in the embedded TP, there is no need to merge a token in Spec-TP. So, we have a token of the DP in each of the relevant positions, necessary to generate drastic interface effects at LF. However, as things are right now, the interpretation procedure (explicature building, in terms of Relevance Theory) will take each token as a separate object, unless we provide an explicit algorithm for *token-collapse*, that is, a procedure the system can use to unify all the tokens for interpretation purposes (both at PF and LF, but our focus will be set in LF). Just as Nunes (2004)

needs an operation Form Chain and then Chain Reduction for PF purposes, we also need an algorithm that not only links all the tokens of a type but also assures complete interpretation. Let us try to sketch such an algorithm, after some definitions, which will be particularly useful when dealing with the data:

56) *Token-collapse*

> Be S a set {α, β…*n*} of arbitrarily complex tokens in positions *P* in a workspace W. An Interface Level *IL* will establish a dependency between the members of S *iff*:
>
> a. The members of S are mutually disconnected
> b. No two members of S belong to the same domain D, and there is no syntactic object SO, such that SO ∈ D and SO is logically equivalent to a member of S for interface purposes
> c. The members of S are structurally identical as far as format is concerned

The reader may have noticed that we have captured the type-token relation in both a formal sense and at the semantic interface, and also accounted for *selection* via Conservation Principle, without resorting to any new assumption, at least within the present framework. Numerations, strictly defined, require numerical subindexes, which are actually *diacritics*, whose interface justification is hard to find. Actually, once one has a natural way to capture *types*, *tokens* follow naturally without stipulations in a Free-Merge framework, interface-driven.

There must be, apparently, some connection between phases, and that connection would be the role of "phase edges" (Boeckx, 2010; Gallego, 2010). However, that only makes sense if one posits that phases are *endocentric* and that labels exist in the syntax. If we do not distinguish between X' and XP, there is no point in talking about edges (i.e., Specs.), at least in the traditional sense. Phase edges are said to have relevance to reconstruction processes, that is, the system can trace back the derivational path of a certain element by looking at its previous positions in the periphery of the phases it had to move through to get to its final destination. In my model, however, things are analyzed differently. To begin with, our phases are not endocentric, but defined in terms of *convergence*, as the ***minimal term***

in a certain level fully interpretable by the next component, following the derivational dynamics presented in (18). If we combine this with a radically "bare phrase structure" (Krivochen, 2011), what results is a picture in which there is neither real "phrase structure" (X-bar theory) nor "phase structure" (Boeckx, 2008), but only *structure* in the underspecified sense of Lasnik & Uriagereka (2012). That is, *there are no phrases in the syntax, there is no projection or labeling either* (which is the expected scenario if we take the hypothesis that the syntactic component is just *generative* seriously), *only free applications of Merge and Transfer of fully interpretable units*. The role of edged and their very existence only make sense if there is something that is *not an edge* (a head or a complement), and that is simply absurd in our framework. Of course there are reconstruction effects at the interface levels, and, in fact, dependencies across phases are *only relevant at the interface levels*, since those dependencies are interpretative, and syntax is a "blind" generative component. Let us put this straight, following much work in "locality-as-(absence of) intervention" (Rizzi, 1990, 2004, 2009 and much related work) but with the focus set on the interfaces, as we have claimed in Krivochen (2012) and Krivochen & Kosta (2013):

57) A *dependency* is *Local* if and only if there is no intervenient object γ (of arbitrary complexity) such that: (i) the relation between α and γ is equivalent to that between α and β *for interface purposes* (ii) α, β and γ belong to the same W and (iii) γ is structurally closer to α than β

Therefore, the only requirement for reconstruction to take place is to give the interpretative component some clue that what is called a *chain* CH = $(X_i \ldots h_i)$ must actually be expressed in terms of *local dependencies between tokens at the relevant interface level*. Movement and copy erasing are thus expressible in terms of *multiple occurrences tokens of the same type* of an element in a *type-array*, motivated by "drastic interface effects", that is, effects in the *explicature*. This way a principled explanation for the Spell-Out of *only one* of the tokens, in standard cases, can be proposed: if an element is displaced, the index (in traditional binding terms) should be maintained across the

derivational path, and the simplest way to do this is by materializing only the copy whose structural position leads the system to optimal relevance.

If an element is Spelled-Out, that is apparently more costly than leaving it as a null copy (not in terms of computation, but in more concrete terms of "linguistic machinery"). The generalization would be "if you can leave something covert, do so. If you make it overt, then you must have a powerful reason for that (e.g., you want to generate some positive cognitive effect that could not have been generated with the covert option)", that reason being very much in the spirit of Grohmann's (2003, 2004) *Condition on Domain Exclusivity*'s "drastic effect on the output":

> *For a given Prolific Domain ΠΔ, an object O in the phrase-marker must receive an exclusive interpretation at the interfaces, unless duplicity of O yields a drastic effect on the output of that ΠΔ.* (2003: 91)

However, even if *all* tokens are Spelled-Out, the interpretative system looks for an interpretation, since there is a presumption of Optimal Relevance (Wilson & Sperber, 2003: 256) that makes the system analyze all the possible interpretations serially until relevance expectations are fulfilled. Copy-erasing would be a clue, like many others (e.g., procedural categories), that leads the inferential system to the intended interpretation. Of course, different languages posit different requirements when it comes to Optimal Relevance at PF, accordingly, the materialization of more than one copy could be required (see, for example, the examples in Nunes, 2004: 38 ff., who attributes the realization of *n* copies to LCA effects). This is not banned by our system; we just state the economy condition that *the minimal number of copies required by PF to achieve Optimal Relevance should be materialized*. This limits the number "on the bottom" to one (since, if no token is to be spelled-out, then it contributes in nothing to generating interface effects, thus violating DFI *and* ConsP, not to mention Minimal Selection), but puts no *a priori* condition on the upper bound. The focus of our proposal is to justify the number of

tokens present at LF as the maximal-minimal number of tokens which convey CS respecting DFI and ConsP. We hope to have provided enough evidence to interest the reader in our proposal.

4. Conclusion:

In this paper we tried to analyze Movement in three levels: *description, explanation* and *justification*. Whether we have been successful or not, is still to be proven. In this conclusion, we will summarize the main ideas of the paper:

i) All operations are interface-driven, and *every derivational step* must be justified in terms of generating a legible representation (*Dynamic Full Interpretation*). The generative operation, *concatenation*, is underspecified enough to apply in any mental faculty, and it is not "designed" in any way to meet interface requirements. The operation must not meet interface requirements, a derivation must. Therefore, DFI (for example) is a condition that applies to derivational steps derived *via concatenation*, but not to *concatenation* as an operation.

ii) Movement, in terms of "interpreting something in a different place from which it is phonetically realized" (as opposed to movement interpreted literally as "displacement of a SO") is triggered by interface reasons, there being no difference with "External Merge". All elements that enter the syntactic workspace are drawn from a *type-array*, in a similar vein as Stroik (2009), but without appealing to features to make elements "survive": *types* "survive" in the relevant sense because they are not used themselves, but instantiated as tokens. In a word, *any (re-)Merge is External Merge* from the *Array*. The division between Internal and External Merge is thus eliminated.

iii) We state the pre-eminence of the semantic component over the phonological component in determining the path the derivation will follow via DFI and ConsP, assuming there is a

pre-linguistic (but nevertheless syntactic) CS whose information is to be linguistically instantiated with a minimum of entropy: the CS determines which elements are to be selected from the Lexicon to derive a linguistic expression that conveys CS. The generative engine needs semantics to have something to manipulate (as Immanuel Kant points out, in his Critique of Pure Reason, "*Gedanken ohne Inhalt sind leer, Anschauungen ohne Begriffe sind blind*"), but can perfectly dispense with phonology, as externalization is not a *sine qua non* condition for syntactic manipulation of symbolic objects. However, the combination might be affected by the availability of phonological exponents to materialize a certain structure. This creates a tension, inherent to NLs, between semantic requirements and phonological possibilities. The possible resolutions to this tension are explored in Uriagereka (2012) and Krivochen (in preparation), with opposite but complementary views.

5. References:

Abels, K. (2003) Successive Cyclicity, Anti-locality and Adposition Stranding. Ph.D. Thesis. University of Connecticut, Storrs.

(2012) *Phases: An essay on cyclicity*. Berlin: De Gruyter.

Alexopulou, T. & F. Keller (2003) Linguistic Complexity, Locality and Resumption. In *WCCFL 22 Proceedings*, ed. G. Garding and M. Tsujimura. Somerville, MA: Cascadilla Press.

Behme, C. (2012) A Potpourri of Chomskyan Science. Retrieved on 10/2/2013 from http://ling.auf.net/lingbuzz/001592

Boeckx, C. (2006) *Linguistic Minimalism: Origins, Concepts, Methods and Aims*. OUP.

(2008) Treelets, not Trees. Talk presented at *BCGL 3 — Trees and Beyond*. May 21–23, 2008

(2009) The Nature of Merge. Consequences for Language, Mind and Biology. In Piatelli Palmarini et. al. *Of Minds and Language*. Oxford, OUP. 44-57.

(2010) Defeating Lexicocentrism. lingBuzz/001130

Bošković, Ž. (2007) On the locality and motivation of Move and Agree: An even more minimal theory. *Linguistic Inquiry* 38: 589-644.

Bošković, Ž. & J. Nunes (2007) The copy theory of movement: A view from PF. In Corver, N. & J. Nunes (Eds.) (2007) *The Copy Theory of Movement*. Amsterdam: John Benjamins. 13-74.

Bresnan, J. (1971) Sentence Stress and Syntactic Transformations. *Language* 47.2: 257–81.

Chomsky, N. (1957) *Syntactic Structures*. The Hague: Mouton.

(1965) *Aspects of the Theory of Syntax*. Cambridge, Mass.: MIT Press.

(1972) *Studies in Semantics in Generative Grammar*, The Hague: Mouton.

(1973) "Conditions on transformations". In *A festschrift for Morris Halle.* Stephen Anderson & Paul Kiparsky (eds), 232–286. New York: Holt, Rinehart and Winston.

(1977) "On Wh-Movement." in Peter Culicover, Thomas Wasow, and Adrian

Akmajian, eds. *Formal Syntax*. pp. 71–132. New York: Academic Press.

(1981) *Lectures on Government and Binding*, Dordrecht, Foris.

(1982) *Some Concepts and Consequences of the Theory of Government and Binding*. Cambridge, Mass.: MIT Press.

(1986) *Barriers*. Cambridge, Mass. MIT press.

(1995) *The Minimalist Program* Cambridge, Mass. MIT press.

(1999) *Derivation by Phase.* MIT Occasional Papers in Linguistics 18.

(2000) "Minimalist inquiries: the framework," in R. Martin, D. Michaels and J. Uriagereka (eds) *Step by Step: Essays on Minimalist Syntax in Honor of Howard Lasnik*, Cambridge, MA: MIT Press, 89–155.

(2001) Beyond Explanatory Adequacy. Ms. MIT.

(2005) *On Phases*. Ms. MIT.

(2007) Approaching UG from below. Ms. MIT.

(2009) Opening Remarks. In Piatelli Palmarini et. al. *Of Minds and Language*. Oxford, OUP. 13-43.

Chomsky, N. and H. Lasnik (1977) On Filters and Control. *Linguistic Inquiry* 8: 425-504.

Corver, N. & J. Nunes (Eds.) (2007) *The Copy Theory of Movement*. Amsterdam: John Benjamins.

Culicover, P. (2001) Parasitic Gaps: A History. In Culicover, P. & P. Postal (eds.) *Parasitic Gaps*. Cambridge, Mass.: MIT Press. 3-68.

Dubinsky, S. (2007) "Parasitic gaps in restrictive and appositive clauses". In *Proceedings of Israel Association for Theoretical Linguistics* (IATL) 22:1-19. Jerusalem: The Hebrew University of Jerusalem.

Emonds, J. (1970) *Root and Structure Preserving Transformations*. Bloomington: IULC.

Epstein, S. (1999) Un-Principled Syntax and the Derivation of Syntactic Relations. In *Working Minimalism*, S. Epstein, and N. Hornstein, eds. M.I.T. Press, Cambridge, MA.

Epstein, S. & T. D. Seely (2002) Rule Applications as Cycles in a Level Free Syntax. In *Derivation and Explanation in the Minimalist Program*, ed. S.D. Epstein & T.D. Seeley, 65-89. Oxford: Blackwell.

Frampton, J. & S. Gutmann. 2002. Crash-proof syntax. In *Derivation and explanation in the Minimalist Program*, S. Epstein & T.D. Seely (eds.), 90–105. Oxford: Blackwell.

Gallego, A. (2010) *Phase Theory*. Amsterdam, John Benjamins.

Green, G. (2011) Elementary Principles of HPSG. In Borsley, R. & K. Börjars (Eds.) *Non-Transformational Syntax. Formal and Explicit Models of Grammar*. London, Blackwell. 9-53.

Grimshaw, J. (1997) Projection, heads, and optimality. *Linguistic Inquiry 28:*373-422.

Grohmann, K. K. (2003) "Prolific Domains. On the Anti-Locality of Movement Dependencies". Amsterdam: John Benjamins.

(2004) Prolific Domains in the Computational System. *In Actas de JEL 2004: Domains*. Nantes: AAI, 211-216.

Gutiérrez Rexach, J. (2000) "The Formal Semantics of Clitic Doubling". In *Journal of Semantics* 16:4, 315-380.

Haegeman, L. & J. Gueron (1999) *English Grammar: A Generative Perspective*. Oxford, Blackwell.

Hale, K. & J. Kayser (2002) *Prolegomena to a Theory of Argument Structure*. Cambridge, Mass.: MIT Press.

Hornstein (2009) *A Theory of Syntax: Minimal Operations and Universal Grammar*. Cambridge, Mass.: CUP.

Jaeggli, O. and K. Safir (Eds.) (1989) *The Null Subject Parameter*. Dordrecht; Boston: Kluwer Academic Publishers.

Jackendoff, R.(1983) *Semantics and Cognition*. Cambridge, MIT Press.

(1990) *Semantic Structures*. Cambridge, MIT Press.

(1997) *The Architecture of the Language Faculty*. Cambridge, Mass. MIT Press.

(2002) *Foundations of Language*. OUP.

Katz, J. and P. M. Postal (1991) Realism vs. conceptualism in linguistics. *Linguistics and Philosophy* 14: 515-554.

Kayne, R. (1994) *The Antisymmetry of Syntax*. Cambridge, Mass. MIT Press.

Kitahara, H. (1997) *Elementary Operations and Optimal Derivations*. Linguistic Inquiry Monographs 31. MIT Press.

Krivochen, D. (2011) An Introduction to Radical Minimalism I: on Merge and Agree. In *IBERIA* vol. 3, 2 pp. 20-65.

(2012) *The Syntax and Semantics of the Nominal Construction*. Potsdam Linguistic Investigations 8, Ed. Peter Kosta et. al. Frankfurt am Main, Peter Lang.

(in preparation) Unlimited Syntax. Ms. UNLP.

Krivochen, D. & P. Kosta (2013) *Eliminating Empty Categories: A Radically Minimalist View on their Ontology and Justification*. Potsdam Linguistic Investigations 11, Ed. Peter Kosta et. al. Frankfurt am Main, Peter Lang.

Larson, B. & N. Hornstein (2012) Copies and Occurrences. Ms. lingbuzz/001484

Lasnik, H. (2011) "What Kind of Computing Device is the Human Language Faculty?". In Di Sciullo, A-M. & C. Boeckx (Eds.) *The Biolinguistic Enterprise*. Oxford: OUP. 354-65.

Lasnik, H. & J. Uriagereka (2012) "Structure". In R. Kempson, T. Fernando, and N. Asher (eds.) *Handbook of Philosophy of Science Volume 14: Philosophy of Linguistics*. Elsevier, pp. 33-61.

Lasnik, H; J. Uriagereka & C. Boeckx (2005) *A Course in Minimalist Syntax*. Oxford, Blackwell.

Mateu Fontanals, J. (2002) *Argument Structure. Relational Construal at the Syntax-Semantics Interface*. Dissertation. Bellaterra: UAB. Retrieved from http://www.tesisenxarxa.net/TDX-1021103-173806/

(2005) Impossible Primitives. In *The Compositionality of Meaning and Content: Foundational Issues*, M. Werning et al. (eds), 213–229. Frankfurt: Ontos.

Martin, R. & J. Uriagereka (2011) On the Nature of Chains in Minimalism. Talk delivered at the conference *Minimalist Program: Quo Vadis?* Universität Potsdam, October 3-6, 2011.

McCawley, J. D. (1968a) The role of semantics in a grammar. In *Universals of Language*,

E. Bach and R.T. Harms (eds.) Holt, Rinehart and Winston.

(1968b) Lexical insertion in a transformational grammar without deep structure. *Papers from the Fourth Regional Meeting.* Chicago Linguistic Society.

Moro, A. (2000). *Dynamic Antisymmetry*. *Movement as a Symmetry Breaking Phenomenon*, Cambridge (Mass.): MIT Press.

Müller, G. (2011) *Constraints on Displacement: A Phase-Based Approach*. Amsterdam, John Benjamins.

Nissenbaum, J. (1998) Movement and derived predicates: evidence from parasitic gaps. In O. Percus and U. Sauerland (eds.), *The Interpretive Tract*, MIT Working Papers in Linguistics 25, 1998.

(2000) Covert Movement and Parasitic Gaps. In M. Hirotani, *et al.* (eds.), *Proceedings of NELS 30*, Amherst, MA: GLSA Publications. 541-555.

Pesetsky, D. (1995) *Zero Syntax: Experiencers and Cascades*. Cambridge, Mass.: MIT Press.

Postal, P. M. (1994) Parasitic and pseudoparasitic gaps. *Linguistic Inquiry* 25, 63-117.

Putnam, M. (2011) The Thing that Should not Be: Rethinking the A-A' distinc-tion. Universitet i Tromso CASTL Linguistics Colloquium, October 7, 2010.

Putnam, M. & T. Stroik (2009) Traveling without moving: The conceptual necessity of Survive-minimalism. In Putnam, M. (ed.) *Towards a Derivational Syntax: Survive Minimalism*. Amsterdam, John Benjamins. 3-20.

Rizzi, L. (1990) *Relativized Minimality*. Cambridge, Mass.: MIT Press.

(2004) Locality and Left Periphery, in A. Belletti, ed., *Structures and Beyond – The Cartography of Syntactic Structures, Vol 3*. 223-251. Oxford University Press.

(2006) On the Form of Chains: Criterial Positions and ECP Effects. In L. Cheng, N. Corver, eds, *Wh-Movement: Moving on*. Cambridge, Mass. MIT Press.

(2009) Movement and Concepts of Locality. In Piatelli Palmarini et. al. *Of Minds and Language*. Oxford, OUP.155-168.

Ross, J. R. (1967) *Constraints on Variables in Syntax*. PhD Diss. MIT.

Sperber, D. & D. Wilson (1986) *Relevance: Communication and Cognition*. Oxford. Blackwell.

Starke, M. (2001) *Move dissolves into Merge*. PhD Thesis. University of Geneva.

Stroik, T. (2009) *Locality in Minimalist Syntax*. Cambridge, Mass.: MIT Press.

Stowell, T. (1981) *Origins of Phrase Structure*. PhD Thesis, MIT.

Talmy, L. (1983) "How language structures space". In Herbert L. Pick, Jr. & Linda P. Acredolo (eds.) *Spatial orientation: Theory, research, and application*. New York: Plenum Press. 225-282.

(2000) *Toward a cognitive semantics*. Cambridge, MA: Massachusetts Institute of Technology.

Uriagereka, J. (1998) *Rhyme and Reason*. MIT Press.

(1999) Multiple Spell-Out. In N. Hornstein & S. Epstein (eds.), *Working Minimalism*, Cambdridge (Mass.), MIT Press, 251-282.

(2002) Multiple Spell-Out. In Uriagereka, ed. *Derivations: Exploring the Dynamics of Syntax*. London, Routledge.

(2012) *Spell-Out and the Minimalist Program*. Oxford: OUP.

Vicente, L. (2009) A Note on the Copy vs. Multidominance Theories of Movement. In *Catalan Journal of Linguistics 8*: 1-23.

Wiland, B. (2009) “Feature valuation by sideward movement”. In Grohmann, K. K. (ed.) *Explorations of Phase Theory*: *Features and Arguments*. Mouton de Gruyter. 53-86.

Wilson, D. & D. Sperber (2003) Relevance Theory. In L. Horn and G. Ward (Eds.) *Handbook of Pragmatics*. Oxford: Blackwell. 607-628.