Revisiting the Evaluator: Derivations (and Learning) in OT

Diego Gabriel Krivochen, University of Potsdam

Institut für Slavistik

Email: diegokrivochen@hotmail.com

Abstract:

This theoretical, programmatic paper has as its purpose to open new perspectives for a crash-proof version of Optimality Theory, in which the GEN function only produces an optimal candidate *per* derivational cycle. We will discuss the concept of *crash-proof syntax* (Putman, 2010), trying to incorporate it into a learning model for neural networks, briefly getting into the *digital computers-quantum computers* debate, comparing the consequences the adoption of one and the other would have for our theory. This objective will be pursued with Minimalism as a *program* -not as a theory[1]- and the theoretical substance will be provided by OT and connectionist models of neural networks. A secondary aim of this paper is to show that mathematical or physical formalizations of natural language do not necessarily imply a "metaphor", but it is possible to work with the hypothesis that natural objects *are in themselves* mathematical structures. Such formalization would be the first step in order to allow a more fluent interdisciplinary scientific exchange between linguistics and formal sciences.

Keywords: Optimality Theory; Radical Minimalism; Neural Network Learning; Quantum Computer

[1] This statement is crucial. Notice that a program does not make substantive claims, only methodological ones (see, for example, Lakatos (1978) for the definition of Program of Scientific Research). From Chomsky (1995) on, there have been substantive assumptions, some of which have no empirical grounds, nor theoretical justification. For example, Epstein & Seely (2002: 65) claim that "*A well-known assumption of Minimalist syntactic inquiry is that sound and meaning are ineliminable, and correspondingly the two interface levels of PF and LF, and only these, are postulated*", and that "*Lexical items, each composed of a bundle of features, also seem ineliminable. Some features of lexical items are illegitimate at one or the other interface*." Notice that the authors are assuming not only an architecture (which is an acceptable claim for a program) but also the notion of feature, interpretability, elimination, lexical item and interface conditions, none of which is in fact a primitive notion. Chomsky's recent claims (2005, 2007) aim at the same scenario, and thus correspond to a Minimalist Theory within the Program, but not to programmatic claims themselves. In the methodological side, "minimalism" is not new, it has been used in Physics and Mathematics (and related branches, like Mathematical Logics) for centuries (see F. Bacon's *Novum Organon*, or Russell & Whitehead's *Principia Mathematica*, for example).

1. Introductory considerations:

To begin with, we will introduce the basic framework within which we will conduct our inquiry. Radical Minimalism is a *program* which analyses language as a fundamentally physical system. Since language is part of the natural world, as Chomsky and his followers have claimed over the years[2], it is assumed to be ruled by the same principles. The program strives to answer questions concerning the integration of what we refer to as *language* in a system of interacting, non-specific cognitive capacities and furthermore advocates that language as a *physical system* should therefore not be studied in isolation, but rather in interaction with other systems. As this interaction occurs in the so-called 'natural world', it is constrained by physical laws which are in turn particular instantiations of mathematical possibilities, particularly within a quantum mechanics framework. Considering this scenario, Radical Minimalism proposes the following tenets (a-d):

(1) a. Language is part of the 'natural world; therefore, it is fundamentally a physical system.

b. As a consequence of (a), it shares the basic properties of physical systems and the same principles can be applied, the only difference being the properties of the elements that are manipulated in the relevant system[3].

c. The operations are taken to be very basic, simple and universal, as well as the constraints upon them, which are determined only by the interaction with other systems.

d. Points (b) and (c) can be summarized as follows:

Strong Radically Minimalist Thesis (SRMT):

> All differences between physical systems are 'superficial' and rely only on the characteristics of their basic units [i.e., the elements that are manipulated], which require minimal adjustments in the formulation of operations and constraints [that is, only notational issues]. At a principled level, ***all physical systems are identical***, make use of the same operations and respond to the same principles.

[2] A reviewer has pointed out that the difference between i- and e- language would make this claim false. However, both i- and e- language have physical reality (mental and external respectively), consequently, they would both be part of the natural world in a trivial sense. In a non-trivial sense, i-language has been taken as the only interesting object for scientific inquiry (as opposed to e-language, competence, etc.), and it is both mentally represented and neurologically based. In a non-trivial sense, then, i-language is part of the natural world, which legitimizes the so-called "biolinguistic" enterprise.

[3] Such a pretention of universality regarding physical explanation is somehow reinforced by the advances in M-Theory –the theory that would unify all variants of string theory (subsuming standard models) and thus constitute a full model of the Universe- and, more humbly, on quantum theory and field theory (see Greene, 1999: Chap. 12, among others). At this point, however, it is a desideratum, but one that, if pursued seriously, could lead to significative improvements in our understanding of the relations between physical systems.

SRMT licenses the possibility of looking for biological, physical and mathematical properties of mental computations (i.e., syntax), *without its being a metaphor* but a true account in three levels: description, explanation and justification. This claim is essential for our argument: we are not saying that language "is like" a physical system and, therefore, a mathematical structure. We are saying that language *is* a physical system. As it is, SRMT is nothing that can be proved right or wrong (it is really a methodological premise one can choose to follow or not), and even if it could, it would be only trivially. However, interesting empirical predictions derive from SRMT, and those can be observationally contrasted (we are consciously adopting a form of Carnap's 1966 model here) in order to determine the descriptive, explanatory and justificative power of the model. Following Carnap's model, certain observational consequences derive from the axioms of a theory: if there is a verification of the observational consequences (i.e., that the relevant object behaves as predicted under experimental conditions), it is interpreted as an *indirect* verification of the initial axioms, which are perpetually under revision. In our case, a strong structural uniformity hypothesis derives from SRMT, thus leading to the observational prediction that, for example, there is a mathematical ratio underlying physical systems, including biological entities. The findings of Prusinkiewicz & Lindenmayer (1990) of Fibonacci patterns in plant growth (e.g., 1990: 37) and Uriagereka (1998, 2012) of the same patterns in phrase structure and linguistic derivations (e.g., 2012, Chapter 7), work as *indirect* evidence of the deep structural uniformity SRMT predicts.

Going back to the levels of adequacy, the description is 'the what', the explanation is 'the how', and finally, the justification is 'the why'[4]. The latter has been either taken for granted or done in a truly non-minimalist way both substantively and methodologically (e.g., triggering operations by means of *ad hoc* features). Our effort, then, will focus on trying to set a radically minimalist alternative of justification, taking into account that a theory of the physical universe must address all three: in this methodological and substantive integration of the study of language within the more general study of physical objects lies the main difference between Radical Minimalism and other approaches. Attempting justification is what we understand as the ultimate goal of going 'beyond explanatory adequacy'.

In the next section we will introduce Optimality Theory, which is the theory we will try to traduce to Radically Minimalist terms, eliminating as many stipulations as possible, while unambiguously formalizing our theorems, as we have done in Krivochen (2012b) taking set theory as the basic framework. Here, we will focus on Dynamic Full Interpretation, as the expression of the simplest possible EVAL function. Then, our focus will be

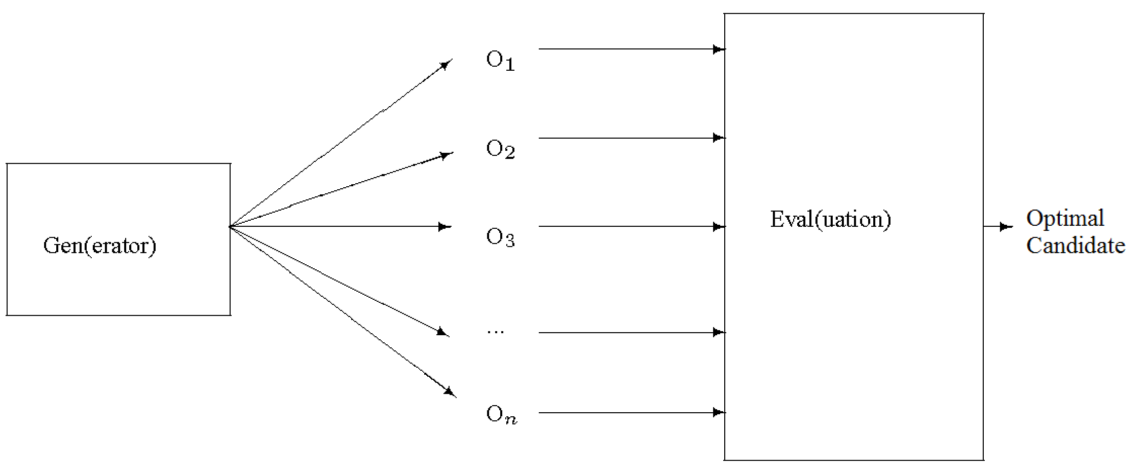
[4] For example, take a Wh- interrogative structure in English. What is observable is that an element appears in a certain structural position, and what is understood is that that position is different from that in which the element is interpreted (the so-called *displacement* property of natural language). That, however, is purely descriptive. Generativism has proposed that the *explanation* of such phenomenon is to be found in the concept of *movement* of constituents, triggered by featural requirements. However, even accepting feature-driven operations we still do not know what physic-mathematical properties of the system or its physical substratum (the configuration of the mind-brain, in an interdisciplinary work with psycholinguistics and neurolinguistics) license the relevant operation (in this case, displacement understood as literal movement of a constituent). This last step is what *justificative adequacy* aims at.

set on problematizing constraint acquisition in an OT grammar, and the advantages a RM approach could bring to shed light on this problem.

2. On the Nature and Form of Constraints:

Optimality Theory (Prince & Smolensky, 1993/2004; Kager, 1999; Barbosa, et. al., 1998) is a theory of the architecture of the mental grammar that is based on the idea that: (i) there are conditions upon output representations, (ii) those conditions (or “constraints”, as they are commonly referred to) are violable to a certain extent, and (iii) those conditions are not universally articulated into “modules” (as in the GB theory) but organized in sets and following a ranking that depends on the relevant language (which accounts for language diversity in areas so different like phonology and syntax). Those constraints were first developed for phonology, and even though the theory has been expanded to all other domains of the grammar, particularly *morpho-syntax*, the basic functions remain the same: the GEN produces a reference set of output candidates ($O_1 \ldots O_n$) to be evaluated by a set of ranked constraints which determine the *optimal candidate*, that candidate belonging to the reference set which minimally violates the set of constraints, provided that all other members of the reference set have a “worse score” on the highest-ranked constraint. This system filters out all candidates, crucially, *but one*. The architecture that results from such a theory has the following form (adapted from Müller, 2011b):

(2)



The status of GEN is not quite controversial, should one assume we are dealing with a version of Minimalist *Merge* operation (see Wartena, 2000; Vogel, 2006; Broekhuis & Vogel, 2010). Contrarily to proposals like Pesetsky & Torrego’s (2007), which require α to probe a feature on β in order to merge (the so-called *Vehicle Requirement on Merge*) or Di Sciullo & Isac (2008)’s set-theoretical restriction on Merge involving proper inclusion between feature bundles, free Merge proposals (for example, Chomsky, 1995; Boeckx, 2010a, b; Krivochen, 2011a, 2012a, b; Ott, 2012 among many others) are plausible and widely accepted theories about the generative component. In the present work will not focus our attention on *generation* but rather on *evaluation*.

The form of *Merge* we will assume here ("concatenation") is to the best of our knowledge the simplest possible one: it is free, unbounded, completely blind to the characteristics of the manipulated elements and *n*-ary, and does not entail labeling or any other further theoretical assumption beyond what is required by conceptual necessity to form structure (adapted from Krivochen, 2012a):

(3) C*oncatenation* defines a set of coordinates in *n*-dimensional workspaces W of the form $\{(x, y, z\ldots n) \subset W_X, (x, y, z\ldots n) \subset W_Y, (x, y, z\ldots n) \subset W_n\}$.

The function is *n*-ary, binarity being derived from interface conditions on interpretation on the one hand, and economy conditions on the other: *two* is the minimal-non-trivial number of distinct elements via which we can generate a *drastic interface effect*. This means that merging α and β (both being defined by their coordinates in the conceptual space, quite in the sense of Wittgenstein) is justified insofar as it has an effect on the output representation, as we will make explicit by means of Dynamic Full Interpretation. Even though considerations on the generative power of the system may be useful for further developments, we will assume this definition for the time being (although the problems are indeed discussed in previous works[5]). Quite like Hoeks & Hendriks (2011: 85), we claim that "the GEN component has no linguistic knowledge whatsoever", in our model, because it is not only in charge of linguistic derivations.

Constraints, on the contrary, are quite problematic, from both a theoretical and an empirical point of view. There are three basic characteristics of OT-like constraints:

a) They are ***universal***
b) They are ***ranked***
c) They are ***violable***

(b) and (c) are claimed to be the main difference between *constraints* and *parameters*, since *universality* is a pretention of both OT and orthodox Minimalism. Moreover, all linguistic variation would rely on different rankings, the lexicon playing no role: the differences are thus not in the component that feeds the syntax (as in GB and Minimalism) but *only* at the interfaces (see, for example, Boersma, Dekkers & van de Weijer, 2000: 10). We will revisit the differences between the approaches below.

Constraints can be classified according to the aspect of the relation input-output that it evaluates, which gives us the following classification (also to be revised below):

[5] Some differences with Chomskyan approaches include the absence of binarity in the definition as well as the elimination of labeling from the definition of Merge. Moreover, Chomskyan-related versions of Merge (Pesetsky & Torrego, 2007; Wurmbrand, 2013; Di Sciullo & Isac, 2008, for example) are frequently *asymmetric* in the sense that Merge satisfies the (featural) requirements of one of the terms being merged, whereas our version makes no such assumption. In any event, asymmetry arises at the interface, but not as a central part of the generation procedure.

- *Faithfulness* constraints: include those constraints that prevent symbols to be deleted or added in the course of the derivation (DEP / MAX constraints), as well as ban operations that internally modify a syntactic object (IDENT constraints).

- *Markedness* constraints: require that the output of GEN meets some structural well-formedness requirement (e.g., syllabic structure).

In this section we will directly present some problems with the current ontology of OT *constraints*, and then reflect upon possible ways to maintain the spirit of OT while solving those problems, in some cases, or directly preventing them from arise, in some others. General objections to the constraint-based OT-program as presented in Kager (1999) Prince & Smolensky (2004) and Müller (2011b) include the following (based on Chomsky, 1995; Burzio, 1994 and our own objections):

a) Where do constraints come from? Are they part of UG (as some claim), whatever it turns out to be? How could OT manage a non-existence hypothesis[6], like Radical Minimalism (henceforth, RM) and a strong version of Survive Minimalism, in which the so-called Faculty of Language FL is nothing more than a dynamic (*n*-dimensional) workspace? How are these constraints really represented in the mind, from a biological point of view?

b) Let us assume there is a number X of constraints. Some of these, say, X_{-4} apply to a L1, and a different subset X_{-3} apply to L2. This is a serious problem, since there are constraints that could plausibly apply to a single language (which takes us back to the Extended Standard Theory *Phrase Structure Rules* and *Transformational Rules*, rejected when explanatory adequacy was taken as the objective of linguistic theory and constructions were regarded epiphenomenal). This critic could be avoided if one accepts the claim that constraints are *universal*, which in turn licenses a different critic: what is the difference between a *constraint* and a *parameter*, a condition on the output or the relation input-output, and a component of Universal Grammar, to be set and configured in contact with Primary Linguistic Data, like the availability of null subjects in finite clauses or the overt Wh-movement (more on this below)? Moreover, how can L1 metalinguistically account for the set of constraints that do not apply to it? If a constraint C does not apply to a language L, can C be formulated in terms of L? The answer is not obvious, particularly if interlinguistic variation is on the spotlight. The development of a metalanguage for the explicitation of the formal procedures is thus problematic, as it has also been problematic

[6] A non-existence hypothesis, like the one we defend here and in previous works, posits that there is no Faculty of Language as a specific "mental organ" as Chomsky and related linguists have argued over the years, but that natural language is the result of an unbounded underspecified generative function (which specifies nothing with respect to the characteristics of the manipulated units) being both limited and fed by systems related to phonology and semantics. In our case, we argue for a preeminence of semantics over phonology, but see Uriagereka (2012) for a different view.

for traditional Generative Grammar to formalize UG (the theory of the initial state of FL) and specify its content, even though every language would be the final, relatively stable state of the FL. Despite the obvious relation that should exist between a theory of the initial state and a theory of the final state, in the best scenario a relation of further specification of the possibilities offered by the genotypical UG.

c) Regarding the ontology of the hierarchy, we find the following problem: let us assume that there is a hierarchy between constraints, so that they have the form $C_1 \ldots >> C_2 \ldots >> C_n$. Which is the principle ruling the hierarchy? In syntactic X-bar representations, there are three axioms[7], *endocentrism* (every projection has a head), *binary branching* (every non-terminal node dominates two syntactic objects) and *projection* (every head projects a phrase). Here, there seems to be no ruling principle. If there is not, there is a fundamental problem with the very concept of hierarchy. But, what is worse, if there *is* (even though Prince, 2002, for instance, provides no prospects for such principles), then there is an even bigger problem: X-bar phrase structure can be seen as a set-theoretical phrase structure (see specially Di Sciullo & Isac, 2008; Panagiotidis, 2010). If this is so, that is, if there is a set relation between constraints such that C_x cannot apply unless C_y, which is higher in the hierarchy, then the application of the most specific constraint can appear *instead* of long chains of constraints. This would work in the same way as simplification procedures express the complex feature composition [+ concrete], [+ animate], [+ human] simply as [+ human], since it presupposes (as it requires) the other two[8]. If there is a true hierarchy, things should work this way, and many constraints could be eliminated in favor of the most specific ones, as the satisfaction of some constraints must be obtained in order for some others to arise in EVAL. But the drawback would be that the more specific the constraint, the less likely it be universal. Consequently, *elimination of redundancy*, which would be desirable in any system, is apparently fatal for an OT evaluator.

Our position will be that OT-like constraints, and representational filters in general, are *inductively* abstracted from Primary Linguistic Data in a concrete speech community, thus having no theoretical *explanatory* entity but fulfilling *descriptive* adequacy requirements. As Prince (2002) argues, however, the formal mechanism of OT is to be distinguished from the substantive constraints that individual authors propose. He says:

"*It is the job of substantive theories in linguistics to provide a restricted vocabulary over which constraints/rules/principles are to be stated. The OT formal framework is neutral on this issue; any*

[7] Even though Kayne (1994), among others, have attempted to derive the X-bar structure from more fundamental relations, such attempts have turned out to be as stipulative as the axioms themselves (in Kayne's case, the notion of asymmetrical c-command depends on a restriction of phrase structure, mainly, bi-dimensionality).

[8] In formal terms, we can say that if $[F_1] >> \{[F2],\ldots[Fn]\}$ (a set of features / constraints), then $\{[F2],\ldots[Fn]\} = [F1]$, or simply create a rewrite rule that allows us to rewrite a set $S = \{[F2],\ldots[Fn]\}$ as [F1].

inadequate restrictiveness of constraints/rules/ principles is a failure of substantive theories, not of the formal framework" (2002: 3)

This comment is of extreme relevance: it allows us to take advantage of the formal mechanisms of OT, which has devoted much attention to the explicit formulation of (interface) constraints, in contrast to the heavily syntacticocentric explanations provided by orthodox Minimalism by means of *ad hoc* feature checking operations. The GEN-EVAL dynamics, as we will see, is indeed a very powerful computational resource, particularly if implemented in a hardware that allows operations beyond Turing-computability.

Objections have arisen to OT as to its psychological plausibility. A *crash-rife* interpretation of the system, as the one Prince (2002: 2) acknowledges, will result in the objection that GEN typically produces an infinite number of candidates to be evaluated, therefore, the theory has no bio-psychological plausibility. While Prince's arguments regarding *performance theories* and *competence theories* (only the former of which would be interested in psychological plausibility, though not in the formal procedures that are made explicit in a competence theory) is not convincing from our point of view, a strictly local interpretation of the EVAL system can save OT from such objections. This local system allows EVAL to access every point of the derivational procedure (in the strongest interpretation) and apply in real time, thus deriving only one candidate, which is at the same time the optimal candidate. Alternatives in this sense have been proposed (McCarthy, 2009; Heck & Müller, 2007), and we will pursue the same line: given an EVAL procedure E, E must apply at every point of a derivation, ensuring well-formedness before proceeding to the next step. We will go back to this, since locality conditions have shown to be essential to linguistic computations: we will formulate a *harmonic serial* function based exclusively on interface requirements.

Another important point is that no constraint system can be fixed: an *n* number of constraints cannot be static, in the sense that it is not likely that the same number of constraints is necessary in every point of brain maturation: this will be the founding stone of our conception of learning in an OT-like neural network, in combination with architectural claims regarding the nature of computational procedures in the human mind (we will review Tesar & Smolensky's 2000 hypothesis, and its compatibility with a *crash-proof* derivational model like the one we will argue in favor of here). For the time being, we will revise the status of constraints and briefly revisit Müller's (2011b) taxonomy of constraints, in the search of a simpler model for OT syntax.

2.2 *Different kinds of constraints?*

In this section we will analyze a taxonomy that underlies most current syntactic research, and was made explicit in Müller (2011a): filters are sorted out according to "*varying degrees of complexity*". We will present each category and the corresponding objections, before giving our own transmodular definition of *complexity* and its importance in syntactic theory.

- *Derivational Constraints:* apply to Merge / Move operations.

We have already defined Merge in a mathematical way, such that it is not constrained by any "featural" condition, as it seems to be the optimal scenario (Cf. Pesetsky & Torrego, 2007 and related work). In other words:

> "***Merge** is a **free unbounded operation** that applies to an n-number of objects sharing format, either ontological or structural*[9]."
>
> Taken from Krivochen (2012a: 5)

A constraint cannot possibly apply to an *operation*, but to its *output*: there is simply no way to evaluate or optimize an operation which is free by nature. The very same example quoted by Müller supports our point of view:

(4) Subjacency Condition (Chomsky (1977)):

In a structure α ... [$_β$ ... [$_γ$ ... δ ...] ...] ..., movement of δ to α cannot apply if β and γ are bounding nodes.

The very formulation of the constraint is clear enough as a counterexample: a determined representation is not suitable as input for the application of a transformational-structure building rule (or, in more general terms, the establishment of a dependency). This is not a constraint over Move, but over the *input* of the operation.

Moreover, even if the aforementioned operation were somehow constrained (in a physical system X), a constraint could assess whether those constraints are met, which are different from the operation itself. Therefore, our position will be that *there are no so-called derivational constraints*. Constraints evaluate *objects*, and *derivations* are ***not*** objects (they are, rather, the record of the series of operations necessary to create and object), but *representations* are.

- *Representational Constraints*: apply to an output operation.

Our objection here has to do with the number of constraints and the learning algorithm that is required. We will posit only *one* constraint in the mature state of the grammatical system, which applies in an extremely local way, being *Transfer* determined by interface conditions, which are *input* constraints (determined by *Interpretative Systems*) rather than *output* constraints.

- *Global Constraints:* apply to a whole derivation.

[9] Following and extending the proposal in Uriagereka (1999), we define *ontological format* as the nature of the elements, which allows *Monotonic Merge*: we do not apply Merge to a root and a number, but to two linguistic elements. On the other hand, *Non-monotonic Merge* involves syntactic objects with the same structural format (in Uriagereka's proposal, two binary-branched tree structures).

These kind of constraints (e.g., the so-called (Generalized) Minimal Link Condition (G)MLC) are simply unformulable in a *phase-driven framework*, and the reason is the following: if constraints are said to apply in the syntax, and syntactic objects are transferred by phase, there is no possible way in which a "whole derivation" is present *at once* in the generative workspace. In any case, *global constraints* should apply *at the interfaces* after the further application of a reconstruction algorithm, which is not what OT has proposed, to the best of our knowledge.

- *Translocal Constraints*: apply to a set of competing output representations to pick out an optimal representation.

This is the classic OT-like constraint, which filters out suboptimal representations. Translocal constraints are traceable to Chomsky's (1957) meta-theoretical desideratum over linguistic theory: it must provide an *evaluation procedure for competing grammars*. In those early days, the criterion was *simplicity*, defined in terms of "*the set of formal properties of grammars that we shall consider in choosing among them*" (Chomsky, 1957: 53). We will redefine *simplicity* for symbolic representations below, as a complement to this more general definition of simplicity as a meta-theoretical requirement. However, in OT there is no such global criterion, which could work very well to rule the constraint hierarchy, and this undermines seriously the very concept of hierarchy. Constraints are proposed and ranked following diverse needs, and new constraints are created *ad hoc* to "demonstrate" an X proposal. This, we will try to get rid of while presenting our Radically Minimalist version of local evaluation.

- *Transderivational Constraints*: apply to sets of derivations, picking out an optimal derivation.

Our objection to this kind of constraints is simple: there are no such things as "sets of derivations" in any derivational point T_X, as interface-driven derivations unfold *in real time*. *Transderivational* constraints are necessarily, if formulable at all, *translocal* constraints. Again, a *derivation* cannot be filtered out, but the resulting representation at a certain derivational point T_X.

These last two types of constraints explicitly express a characteristic of OT-syntax that we would like to revisit here, more specifically in connection to the learning algorithm we will present below: OT *requires* GEN to produce many alternative outputs, thus leading to a *crash-rife* syntactic model in which all candidates are dismissed but one. In a *crash-proof* model (Putnam & Stroik, 2012; Putnam, 2009), GEN should optimally generate *one* candidate, which is in turn an *optimal* candidate. However, is this possible in early stages of (L1) grammar development? We believe not: if this was the case, that is, of early grammar was already crash-proof, there would be no explanation for the fact that language acquisition takes some time. This time, which is not compatible, as Lasnik has pointed out, with quick parameter setting either, we will identify with the time required to adjust neural networks to generate an optimal candidate provided certain output conditions, what in connectionist networks is known as *supervised learning* (Carreiras, 1997). The thresholds for each network must be fixed according to experience (Primary Data, in Chomskyan terms), and it is thus not possible to have a perfect GEN-EVAL interaction from square one. This is why we are strongly against a fixed constraint system, which is not able to show the ontogenetic evolution from a crash-allowing mental grammar to a crash-proof one, this being our understanding of "language acquisition".

2.3 *On Constraints and Parameters:*

We will make a short note on Constraint-based syntax and Principles & Parameters-based syntax. OT-defenders claim that there are strong differences between those two kinds of models, and argue in favor of the former (see Broekhuis, 2006, for example). We will, in turn, argue that there is not much difference between them, at least not enough to distinguish two essentially different programs of research.

Let us review some apparent differences (taken from Kager, 1999 and Müller, 2011b):

- Constraints are *violable*, whereas Principles are *not*

This is not so, especially in the early Principles & Parameters framework. Barrier violations were allowed, and the notion of "*increasing degree of ungrammaticality*" was put forth in direct proportion to the number of crossed bounding nodes / barriers. Even in current models, a principle like the Phase Impenetrability Condition (PIC) is not clear to apply in cases of so-called "sidewards movement" and scrambling (see Kosta, 2006).

- Parameters have two values, whereas Constraints are simple one-sided statements

Let us review a well-known pair of constraints, posited by Müller (2011b), among others:

(5) *WH-CRITERION (WH-CRIT):*
Wh-items are in SpecC[wh].

θ-ASSIGNMENT (θ-ASSIGN):
Internal arguments of V are c-commanded by V.

In our opinion, these so-called "constraints" are nothing more than the two possible settings of the "Wh-parameter" (which determines the obligatoriness of *overt* Wh- movement to the left periphery of the clause, as the specifier of a [+ Q] C_0 head), with the extra elements and relations "Spec-of", "Wh-feature", "c-commanded-by". In this sense, the proposal is even less economical (and thus biologically implausible) than standard P&P approaches. Moreover, there is significative overlapping between θ-ASSIGN and *MOVE (to be found in Kager, 1999 and Grimshaw, 2006; among many others), which favors elements *in situ*.

- Constraints are *ranked*, whereas Parameters are *not*

This is another misunderstanding of recent developments within the P&P framework. Uriagereka (2007) asks whether OT and the MP are compatible or not. Putting aside the question of whether the problem itself is real (if OT is a theory and MP a methodological program, then there should be no problem at all), Uriagereka distinguishes *macro-parameters* from *micro-parameters*. Macro-parameters are early set and determine the *possibility* of setting more specific, possibly *more local*, parameters. These micro-parameters, external to the "core" UG, are subjected to change over time, more easily than macro-parameters, which are closely associated to linguistic typology. There is, then, a rank in current parametrical theory, but the same criticisms apply: which is the principle ruling the hierarchy? In our presentation, we will dispense with both systems as they are conceived now and develop a whole new framework based on Krivochen's (2011a, et. seq.) Radical Minimalism and Putnam's (2010) *crash-proof* proposals. The advantages we find are the following: the elimination of constraints is a theoretical improvement that, if proven correct empirically, leads to a considerable simplification of the OT framework. At the same time, we maintain the basic OT GEN-EVAL architecture, but making the GEN algorithm explicit (without resorting to feature-driven operations but working with a free unbounded GEN) and the EVAL conditions are purely third-factor constraints. Moreover, we will problematize acquisition from an OT point of view, proposing what an ontogenetic development of the EVAL component could look like in a dynamic architecture. We also draw on Putnam's (2010) notion of *soft*- versus *strict-crash* to locally evaluate the products of GEN, thus getting a system with possibilities to define local domains for syntactic operations.
We agree with Broekhius and Vogel (2010: 245) in that "(…) *even when it turns out to be possible to develop a crash-proof syntax that only generates well-formed objects that satisfy the interface conditions, filters on the output of the computational system will remain an essential ingredient of the theory of syntax* (…)", but it is not the case that a crash-proof model can be developed independently from a theory of constraints: the subsumption of constraints to one EVAL condition driving the syntactic GEN is what makes the system crash-proof. We propose, then, a bidirectional flow of information: the EVAL component "asks" the syntax to build an object while respecting its constraint(s) and then locally evaluates the derivation, in order to correct the mechanism, building on connectionist approaches to neural network learning (Elman, 1990). In the next section we will focus on the EVAL component, how to formulate it and how it would look under a Radically Minimalist light.

3. The EVAL(uator) revisited:

In OT, the EVAL component is in charge of filtering out illegitimate representations generated by GEN. That evaluation, in *most* current versions of OT, applies at a global level (see Embick, 2010 for discussion), even though *globalism does not necessarily follow from an OT-like architecture*. The main problem we see here is that *global* evaluation is both computationally inefficient and biologically implausible: biological processes tend to occur in "chunks", as we can see in DNA with the relation between nucleotides and aminoacids. If we consider OT-like constraints in a *local* domain, things get only a little better: the problems with the ontology of the hierarchy remain. In turn, extremely local optimization (Müller, 2011b; Heck & Müller, 2010) leads to a proposal in which *all SO are –stipulatively- phases*, as in the strongest interpretations of Epstein & Seely (2002): bear in mind that the word *optimization* is used, not *evaluation*. If a SO is an optimal candidate, there is no reason why it should not be transferred *ipso facto*, regardless interface legibility conditions: a blind application of Pesetsky's (1995) *Earliness Principle*. Extremely local *optimization* leads to a crash-rife model, in which, as Epstein & Seely (2006, 2007) and Putnam (2010) notice, there is no place for "momentarily crashing" objects, which they refer to as a "strict crash" model. Let us take the following sentence (from Putnam, 2010: 4):

(6) Rats like cheese

The derivation of this sentence involves the creation of the following object in a derivational point T_X,

(7) $[_V$ like $[_D$ cheese]]

which is not interpretable, either in an orthodox Minimalist framework (violation of the Theta-criterion, as there is an external role not assigned, assuming V is capable of assigning Case) or in a Radically Minimalist framework (no fully propositional form can be built if the C-I interface takes that SO from the workspace). Extremely local optimization as presented in Müller (2011b) and Heck & Müller (2010) inevitably leads to crash at almost every point (i.e., partial representation created) in a derivation. This can be better understood if we consider two key assumptions underlying the GB conception of "well-formedness" (taken from Epstein & Seely, 2006: 179):

(8)

a. All and only grammatical sentences have 'well-formed' derivations.
b. A derivation is well-formed only if at every point in the derivation no principle is violated.

The goal of a revised OT (which we will pursue) would be then to model a local *evaluation* algorithm which is compatible with a softer version of *crash-proofness*: at some points in a derivation, violations must be tolerated, at least until the next derivational step and the evaluation of the output. The definition of *crash* that best suits this logic is referred to as "Soft-crash" (Putnam, 2010: 8):

(9) *If a syntactic object α cannot be interpreted at an* [Interface Level] *IL in any and all of its features, α is neither useable nor legible at IL, iff α cannot be combined with another local derivational unit that repairs the violation(s) of α.*

Following these claims, our version of *harmonic serialism*, taking into account bidirectional information flow, is expressed in the following principle, an extremely local (i.e., step-by-step) evaluation measure for derivations:

(10) ***Dynamic (Full) Interpretation***: *any derivational step is justified only insofar as it increases the information and/or it generates an interpretable object.*

The *or* clause allows "soft crashes" like (7), as the merger between [$_V$ like] and [$_D$ cheese] does not generate a fully interpretable object for *both* interfaces, but it increases the informational load, and so avoids entropy (i.e., the loss of information as the system develops in time under normal conditions), which is the main aim of interface-driven generation. The principle we have formulated explicitly allows the following kinds of derivational steps:

(11) a. Increasing the informational load without it resulting in a fully derivational object

b. Increasing the informational load *and* generating a fully interpretable object

We will exemplify those kinds, and it will be obvious that, while it can be the case that the informational load can be increased without generating a fully interpretable object, the inverse is not possible (that is why we have put an *or* connective). Consider the following phrase marker, locally evaluated (we follow Hale & Keyser's 2005 and Mateu Fontanals' 2002, 2010 representations for ditransitive predicates, also used by Bleam, 2001 and related work; in which both internal arguments are arguments of P, not of V *contra* Larson, 1988):

(12) [VP [V] [PP [DP$_1$] [[P] [DP$_2$]]]]

Dative interpretation

The case dimension in DP$_1$ cannot be valued by V, as we need a causative layer that generates the interpretation of "affected object" or a finite T layer if V turns out to be unaccusative to generate a thematic interpretation on the DP$_1$. Let us assume the first scenario, a di-transitive construction. The immediately next derivational step would be (13):

(13) [*v* [VP [V] [PP [DP$_1$] [[P] [DP$_2$]]]]

Accusative interpretation

In (13) the *v* head generates an ACC interpretation for DP$_1$ and *v* adds causativity to the construal. All DPs in the representation now have a Case interpretation, but the phrase marker in (13) is not yet fully interpretable: the *v* head licenses (in fact, requires) an external argument, a *cause* or *initiator*. This argument, however, will not be introduced until the next derivational step. The merger of *v*, therefore, increases the informational load but generates an only *partially* interpretable object. The notion of *soft crash* we have borrowed from Putnam (2010) is essential here, as the system allows momentarily non-convergent elements, under the condition that all nodes introduced so far receive an interpretation at the interfaces (particularly LF). The introduction of *v* is thus an operation of the kind of (11 a). Once the initiator is merged, its Case potentiality must be collapsed by the merger of a finite T node, whose introduction would be an operation of the kind (11 b). EPP effects, which we will not discuss here, can be subsumed to theme-rheme dynamics (as argued for in Krivochen and Kosta, 2013), leaving us with no uninterpretable features blocking interface interpretation.

In this framework, we could even argue that *optimality* is an epiphenomenon or a particular case of this increase in the informational load in a symbolic structure in which a fully interpretable object is built, in any case a *subset* of the syntactic objects generated by Merge with the guidance of DFI. DFI and our dynamic definition of *phase* require *real-time interpretation*, that is, each generated structure is analyzed by the interfaces and if things are going well within tolerable local boundaries (i.e., the *next* derivational step), the derivation continues:

(14) *P is a* ***phase in a system*** $\boldsymbol{S_X}$ *iff it is the* ***minimal term fully interpretable in*** $\boldsymbol{S_{X+1}}$

The "invasive interfaces" take the minimal fully legible SO from the generative workspace *as soon as that object is assembled* (which the relevant interface can know via *analyze*), thus defining a *phase* in dynamic terms: there are no *heads* determining transfer points (in fact, no heads at all, since in an *n*-D syntax, headedness is an epiphenomenon) as in Chomsky's approach, or [u-F] signaling the limits of a transferrable object (as in Gallego's (2010) approach). There is no "every phrase is a phase" (Epstein & Seely, 2006) problem if phases are exclusively determined by interface requirements, and transfer is interpreted as "interfaces taking" rather than "generative workspace sending": phrases are actually epiphenomenal readings of *n*-any concatenations, provided that C-I interface conditions, as we have argued in Krivochen (2011a), restrict *n* to 2 in the case of natural language (but, crucially, not in the case of geometry, particularly elliptical geometry or fractal geometry, for instance).

Continuing with our explicitation of the derivational mechanism we will use, C-I evaluates *every output* of Merge but, crucially, no *optimization* takes place, as we have said before. This version of *harmonic serialism* thus dispenses with the requisite of optimization, as our system is *softly crash-proof* or, in other words, allows momentarily non-legible units. This "invasion" of the interfaces into the working area receives the name of *Analyze*, and it is not in itself an independent operation, but merely a consequence of the architecture of a massively modular conception of the mind and the distinction between generative workspaces, which are dynamic and non-permanent, and interpretative systems, which limit the otherwise unbounded outputs of the geerator. Our derivational dynamics, then, has the following form:

(15) Generative System: *Concatenate* (α, β…*n*) = {α, β…*n*}
Interpretative System(s) –if any-: *Analyze* {α, β…*n*}

Where *Analyze* is the local evaluation operation via which the interpretative systems access the workspace and look for the minimum fully legible object to be *Transferred*:

(16) Given a syntactic object SO, SO is fully legible at the Interface Level IL *iff*

- $\forall(x) \mid x \in \text{SO}, x \in \text{IL}_1, \text{IL}_2, \ldots \text{IL}_n$ (where IL = Phonology, Semantics)

Being IL itself a set of distinctive features, in a Jakobsonian manner. For example,

(17) Phon = {round, frontal, voiced, palatal…}

If (16) applies for a SO, that SO is taken by the relevant interface.

Because it is feature-free and interface-driven, our *local evaluation system* is more flexible than most current proposals (some of which we have already reviewed), allowing the working memory to host structures of variable complexity before *Transfer* applies and the interfaces take what they can minimally read[10]. Given the fact that the Faculty of Language (FL) is in Radically Minimalist terms nothing more than a dynamic *n*-dimensional workspace originated from the intersection of two systems (CI / SM) and the activation of the prefrontal cortex (see D' Espósito, 2007 for a neural model of working memory in these terms, as well as the articles in Arbib, 2003, particularly *Neurolinguistics*, *Layered Computation in Neural Networks* and *Learning network Topology*), it exists *within* those systems, a proposal that is very similar in spirit to Stroik & Putnam's (in press) *Survive Minimalism*. Therefore, it is only natural that the so-called "external systems" (which are not "external" at all in the sense that there is no central language-specific syntactic component to be external to) can have access to the derivational steps, what we call "*invasive interfaces*".

This on-line interpretation may sound more costly than waiting until the whole derivation is finished and only then evaluate it, but a dynamic definition of *phase* requires C-I to determine if a syntactic object is fully interpretable in order to be transferred. Our *evaluation* system will take the name of *Analyze*, a derivational step (not an operation) in which the interfaces evaluate the output of GEN dynamically using as "constraints" their own legibility conditions.

Manipulation of elements is free, but, of course, as certain patterns emerge as *frequently relevant*, those structures are built as the first option to be considered by the semantic component. Therefore, by resorting to adjustment of neural connections (roughly, statistical learning as described, among others, by Thornton & Tesan, 2007), we can account for the generation of convergent structures without stipulations constraining *syntax*, which is a simple, general-purpose concatenation function, the GEN algorithm as formalized in (3). This will acquire major importance when we present *language* as a *chaotic system* in following sections.

[10] This is an important difference with Chomsky's (2005) phase system, in which transferrable units are fixed and defined *a priori*, namely, *v**Ps and CPs.

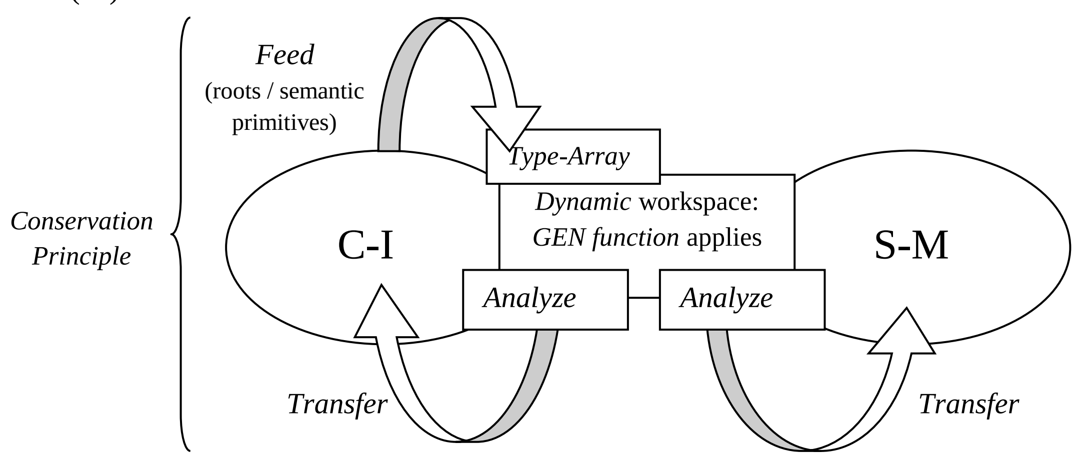
3.1 *How the system works:*

Once we have spelled out our machinery, let us put it in practice with a ditransitive example, following the model of (12) and (13), and the aforementioned references:

(18) [EA [[*v*] [[V] [IA [[P] Loc]]]]

That is, we have a *caused event* (*v*-V relation) which includes a *locative relation* (P) between a theme (IA) and a location (Loc), in terms of terminal coincidence (which could be spelled out by the preposition *to*). Now, where does the information come from? Let us remind the reader of the architecture we have proposed in Krivochen (2012a, b) and Krivochen & Kosta (2013):

(19)



We begin with a pre-linguistic conceptual structure, following the line of Jackendoff (1987, 1997, 2002) and Mateu Fontanals (2002) among others, in accordance with the fodorian tradition of a "language of thought". Such a "language" is taken to be a conceptual structure, generated by Merge applying to generic concepts, pre-linguistic and *a*-categorial (as they do not have category nor the potentiality to bear one). The concepts we are referring to can be found as "primitives" in Jackendoff's work, even though we reckon the fact that most are in fact composite structures involving a syntactic relation between more atomic concepts. Let us assume, as Mateu Fontanals (2002, 2010) does, two kinds of elements within the causative-eventive domain (the θ-domain in Grohmann's 2003 terms): relational elements (linguistically instantiated as P, *v* and V) and non-relational elements (linguistically instantiated as N), roughly identifiable with logical predicates and arguments respectively. Those concepts are structured in C-I following a semantic-pragmatic *intention* (thus the "Intentional" part of C-I, long neglected in orthodox studies), another claim with long history (see Bernárdez, 1983 for details with respect to a semantic-pragmatic *global plan* in text linguistics, for example; or Carreiras, 1997 for a discussion of the connectionist vision of language production, also involving these kind of intentions), conveying *information*. Information cannot be destroyed or created, just transformed[11], a principle Lasnik, Uriagereka & Boeckx (2005) also adapt (although with different purposes, see Krivochen, 2011a for discussion), and we formulate as follows (as a unification of *faithfulness* constraints):

(20) ***Conservation Principle (ConsP)****: information cannot be eliminated in the course of a derivation, but it must be instantiated in the relevant system in such a way that it can be read and it is fully preserved.*

Notice that, in OT terms, this would be a standard formulation of *Faithfulness constraints*, all subsumed to a general principle of physical systems, not language-specific (thus revealing interesting characteristics of both language and other physical systems, if applicable. See Uriagereka, 1998, 2012 for further discussion). Given *ConsP*, we have a further tool to integrate language in the broader perspective of the studies of physical systems, beyond the limits the "biolinguistic enterprise" (see Di Sciullo & Boeckx, 2011 for a recent overview) has set itself with reference to biological systems, without pursuing a deeper level of explanation (with the notable exceptions of Jenkins' and Lasnik's contributions to the aforementioned volume).

Once we have the semantic structure, taking Mateu Fontanals' (2002) terminology, a Relational Semantic Structure (RSS), the information cannot be deleted, just *modified*: this is the *transduction* to the linguistic workspace, where the concepts are *instantiated* as *roots* (or, in Boeckx's 2010b terminology, *conceptual addresses*), and relational elements are instantiated as functional-procedural elements (*v* for [cause], V for [event] and P for [location]). Specifically linguistic nodes are added in this point, T, Aspect, Modality. Bear in mind that, as we will see, every instance of Merge is triggered by DFI, that is, it is *interface-required* to increase the informational load. Thus, our derivation is *not* feature-driven, *contra* Pesetsky & Torrego (2007) and related *constructivist* work (e.g., Lasnik, Uriagereka & Boeckx, 2005; Di Sciullo & Isac, 2008). C-I evaluates every output of Merge, but crucially, no *optimization* takes place, as we have said before. This "invasion" of the interfaces into the working area receives the label of *Analyze*, and it is not in itself an independent operation, but merely a consequence of the architecture of the mind as we see it, in close relation to Survive Minimalism's (Stroik & Putnam, in press)

Prior to deriving the sentence, we will make some comments which we find useful for expository purposes:

- Parallel derivational spaces have been discussed in the literature; see Uriagereka (2002) for details. We will maintain that mechanic for complex-unit Merge. The post-syntactic C-I component, with which we

[11] For example, via Merge. Consider Merge, {α, β} = K. It would be a mistake to claim that K *creates* information, since there can be nothing in K that is not present in α and β. Chomsky's (2001) labeling algorithm serves as a proof: if α = N and β = V, K cannot be labeled P, since the object would be wrongly manipulated in future computations.

identify the optimal EVAL, also appears to be able to work in parallel, for example, when deriving *higher-level explicatures* and *implicatures* at the same time (Wilson & Sperber, 2003).

- We will just build a *complete thematic domain in a ditransitive structure*, leaving the derivation of higher nodes (Mod, Asp, T) to the reader, following the same mechanics.
- Conceptual elements are taken to have a *categorical potential*, that is, they can "collapse" to a category in a certain distribution (i.e., a local relation to a functional node). Thus, categories are not created in the syntax, nor are there categorical features, but they are interface readings of syntactic configurations. We will assume that D collapses the categorial potentiality of an undetermined root to N (i.e., sortal entity) without excluding [cause], T collapses it to V (extending-into-time perspective) and P collapses it into A or Adv, depending on the modified element (see Mateu Fontanals, 2002). Common sense may dictate that the primitive *cause* appears only in verbal (i.e., eventive) structures, but there is an aspect of the C-I/syntax interface that we have mentioned elsewhere and is essential to this: *this interface (and possibly, all other interfaces) is not transparent* (i.e., there is no exact correlation between a Relational Semantic Structure and its syntactic realization, as well as there is no exact isomorphism between the representations manipulated by two modules, even respecting the *ConsP*)[12].
- What we have claimed for category is also extensive to Case: in our framework, Case is nothing more than a morphological epiphenomenon, also a reading of a syntactic configuration (the abstract potentiality being *represented* by the notation [Case_X]), without any entity in a GEN that is purely generative: otherwise, we should claim that GEN is partly interpretative, as in orthodox Minimalism (where the syntactic component has access to values and features). The Case interpretations will follow Krivochen (2012b) ***Case Sphere Theory***, which we very briefly summarize here:

<u>Nominative</u>: *read off* from a {Time, {D}} local relation, and *interpreted thematically* (in the explicature building process, see Sperber & Wilson, 2003) as Agent / Force

<u>Accusative</u>: *read off* from a {Cause, {D}} local relation, and *interpreted thematically* as Theme, the object (Figure) located in / moving towards, etc. a Ground

<u>Dative</u>: *read off* from a {P, {D}} local relation, and *interpreted thematically* as Location, the Ground in Talmy's (2000) terms.

Let us now turn to the derivation itself:

(21) *Array* = {D, P, √α, √β, √γ, *cause, event*}

[12] For example, Krivochen's (2011c) derivation of relational adjectives makes use of *cause* and *event* within nominal structures.

1) *Narrow Syntax* Merge ($D_{[CaseX]}$, √α) = {D, √α}

2) *C-I* Label {$D_{[CaseX]}$, √α} = {D, {$D_{[CaseX]}$, √α}} *This* {D} *will be taken as a unit for the purpose of future operations*. Incidentally, {$D_{[CaseX]}$, √α} "categorizes" √ as N, following our definition.

3) *C-I* Analyze: not fully interpretable unit: D has an uncollapsed potentiality for Case interpretation.

4) *NS* Merge (P, {$D_{[CaseX]}$}) = {P, {$D_{[CaseX]}$}} P*'s procedural instructions collapse* [$Case_X$] *on* {D} to DAT sphere.

5) *C-I* Label {P, {$D_{[DAT]}$}} = {P, {P, {$D_{[DAT]}$}}}

6) *C-I* Analyze: {D}'s referential properties depend on the cumulative influence of Time, Aspect and Modality, if it is a common name. Proper names are taken to be inherently manipulable by C-I (see Krivochen, 2012b). This unit is not fully interpretable yet. Relational element P requires another element (a *figure*).

7) *NS* Merge ($D_{[CaseX]}$, √β) in parallel to (1) = {$D_{[CaseX]}$, √β} Labeling and Analyzing also take place. No procedural head can collapse {D}'s Case potentiality, so the structure is not yet fully interpretable.

8) *NS* Merge by *Structural Format* ({D}, {P, {P, {D}}}) = {{D}, {P, {P, {D}}}}

9) *C-I* Label {{D}, {P, {P, {D}}}} = {P}.

10) *C-I* Analyze: {D} has a [$Case_X$] potentiality still uncollapsed. Not fully interpretable. Therefore, the whole domain P containing {D} is not interpretable either.

11) *NS* Merge ([event], {P}) = {[event], {P}}

12) *C-I* Label {[event], {P}} = {event, {[event], {P}}}

13) *C-I* Analyze: idem (10)

14) *NS* Merge ([cause], {event}) = {[cause], {event}} Procedural instructions on [cause] can collapse [$Case_X$] on the closest {D} structure to ACC sphere.

15) *C-I* Analyze: is {P} now fully interpretable? Let us assume P = *central coincidence* (i.e., [WITH]), which gives the P domain a clausal flavor since the analysis of Double Object Constructions show that P [WITH] is semantically equivalent to V [HAVE]. P is then a fully interpretable object, no potentialities are left unresolved, what we will later on refer to as the "ψ-state" of an interpretable dimension, following a standard notation in quantum mechanics.

16) *NS* Transfer {P}

17) *C-I* Label {[cause], {event}} = {cause, {[cause], {event}}}

18) *C-I* Analyze: two procedural instructions will cause collapse, since there is no conceptual root to provide the "semantic substance" needed for an explicature (that is, a full propositional form) to be built. [cause] licenses an external position for an initiator, forcing the system to "wait one more turn".

19) *NS* Merge (D, √γ) in parallel = {D, √γ}. Idem (8).

20) *NS* Merge by *Structural Format* ({D}, {cause, {[cause], {event}}}) = {{D}, {cause, {[cause], {event}}}}.

21) *C-I* Label {{D}, {cause, {[cause], {event}}}} = {cause, {{D}, {cause, {[cause], {event}}}}}

Because it is feature-free and interface-driven, our *local evaluation system* is more flexible than most current proposals (some of which we have already reviewed), allowing the working memory to host structures of variable complexity before *Transfer* applies and the interfaces take what they can minimally read[13]. However, a further step must be taken, since if the computation is interface-driven, there must be a criterion that, in turn, drives the interfaces' requirements. Within Generative Grammar, this criterion has been "simplicity" since Chomsky (1957). Therefore, it is essential for linguistic theory to have some *formal* definition of *complexity*, since otherwise it is impossible to even start the search for "principles of efficient computation" which, apparently, should prefer *simplicity* over *complexity*. Another issue, with stronger focus on OT-models, is whether constraints are *complexity-sensitive*, that is, if they apply only to objects up to a certain degree of complexity. We will propose a definition that allows a formal characterization of the notion, without circumscribing ourselves to linguistics:

(22) *An object α is more complex (or less simple) than β iff: (a) building of α involves more derivational steps than β (i.e., the application of a further algorithm) (b) α does not imply a further drastic effect on the output over β.*

We define *complexity* only when there is an *interpretative system* involved and taking into account legibility conditions. A direct consequence of this claim is that no definition of *complexity* can be provided for an object *in abstracto*, but only taking into consideration a reference set and the fact that the relevant object will be compared to other/s at some interface level. Thus, for example, a *sentence* is not *a priori* "more complex" than a *word* if the interface effects we want to achieve can be obtained with a sentence but *not* with a word. Just like Relevance,

[13] This is an important difference with Chomsky's (2005) phase system, in which transferrable units are fixed and defined *a priori*, namely, *v**Ps and CPs.

complexity is a notion that is to be defined *at the interfaces* and as a *cost-benefit* relation. And, in the case of sub-personal systems, it can also be defined in biological terms, which are nothing more than specific instantiations of more general mathematical/physical principles. In this sense, our definition differs from that of Jakubowicz (2011: 340) in that it is not the "number" of Merge applications, but the effects those instances of Merge have at the interface levels:

(23) ***Derivational Complexity Metric***

a. Merging α_i *n* times gives rise to a less complex derivation than merging α_i ($n + 1$) times.

b. Internal Merge of α gives rise to a less complex derivation than Internal Merge of $\alpha + \beta$.

As we said, particularly in a free token-Merge system as the one we have outlined here and in past works, if we Merge-α *n* times following DFI, then a derivational point D with *n* instances of α and a derivational point D' with $n + 1$ instances of α may be equally complex, if the interface effects of both derivational points are equal, *ceteris paribus* (that is, if the model in question allows the introduction of elements that do not increase the informational load, like Agr- projections).

Our definition also contrasts with that of Edmonds (1999) insofar as he argues in favor of a *syntactic* measure of complexity, whereas we take a straight *interface* approach to the phenomenon. His approach is, like ours, a comparative one (he defines complexity only in the event that similar systems are considered and a task is to be performed, 1999: 46, ff.), but it has not been designed for natural languages, where there are post-syntactic processes (e.g., LF movement, reconstruction effects at LF, prosody assignment at PF), some allegedly computational, which should also be compared (optimally, using the very same criterion/a). Even though a definition like that proposed by Edmonds might be of use for the analysis of formal languages or the comparison of scientific theories as formal axiomatic systems (since he explicits and develops the criterion of simplicity, way more than Chomsky ever did), it is not enough for *natural* languages, as we have not only a generative operation but also interpretative interfaces (which, in turn, are computational).

The line of reasoning with respect to the place of language within the physical universe we are pursuing is the following: take Tegmark's (2007: 1) *Mathematical Universe Hypothesis*:

(24) **Mathematical Universe Hypothesis (MUH)**

Our external physical reality is a mathematical structure.

A (mathematical) structure is defined as a set S of abstract entities and relations $\{R_1, R_2 \ldots R_n\}$ between them. Essentially, MUH is not restricted to *our* Universe, and in a multiverse theory, this is a very relevant claim. Each Universe, then, is defined as a set of entities and relations, in very much the same way we have defined the basic tenets of Radical Minimalism. In this context, formalization is essential (see Krivochen, 2012a for a mathematical formalization of RM and cf. Collins & Stabler, 2012, for a formalization of orthodox Minimalism).

Now, accepting that MUH is valid for *every possible Universe*, we have to see how physics is affected. A physical claim is a *particular instantiation* in a given Universe of a pair {E, O}, in which E is in itself an entity or set of entities and O the relevant operation upon E. Basically, $R_X = \{E, O\}$ & $R_X \in W_X$, where W is an *n*-dimensional workspace. Physical statements depend on the existence of a Universe and an external physical reality, something a mathematical claim can dispense with. Necessity[14], then, is a matter of mathematics, but not of physics. In Radical Minimalism, *necessity* in this strong sense is embodied in the notion of the *concatenation function*, which can give rise to any kind of hierarchical structure (we repeat the definition for the reader's convenience):

(25) *Concatenation* defines a *chain* of coordinates $\{(x, y, z\ldots n) \subset W_X, \ldots (x, y, z\ldots n) \subset W_Y, \ldots(x, y, z\ldots n) \subset W_n\}$

Accepting the claim that a physical claim is a particular instantiation of a (portion of a) mathematical structure, what is the place of biology within this hierarchy? Our position is that *biology is a particular instantiation of a physical system*, if we take "physical system" in the sense of "specific portion of the Universe taken for analysis". Biological statements are not formal, and we could very well say that biology is the science of the contingent, in the sense that *necessity* in our strong sense is simply unconceivable. Any biological claim is ultimately a mathematical claim but with a higher level of specificity and involving entities in a determined Universe. Thus, Biolinguistics, important though it is, must not be taken as the ultimate field of inquiry.

Following this line, we will attempt at *mathematically* formalize OT-like constraints, and develop a theory of dynamic development that obeys a different learning algorithm, within a framework of quantum computing. This will allow us not only to gain descriptive adequacy, but also reach *justificative adequacy* (Krivochen, 2011a) and hopefully start answering the question of *why the architecture is the way it is*, not only *what* it is like or *how* it works.

4. A learning algorithm for OT-syntax?

In its current standard versions[15], OT is a *static* constraint theory, in which a variable *n* number of constraints (depending on the author) operate upon GEN outputs regardless stage in the ontogenetical development (from an empirical point of view) or *Occam's Razor* (from a theoretical point of view). Acquisition in OT is *constraint ranking*, provided that the set of constraints has been made available by UG (Tesar & Smolensky, 2000). The number of constraints is determined beforehand, as well as their identity (i.e., which specific constraints will be

[14] We understand *necessity* in the strongest possible sense: a proposition is *necessary* if and only if it is true in *every possible Universe*. The claim that complexity derives from some form of a concatenation function seems *necessary* to us in this sense.

[15] We think it is most fair to object to the versions of OT that have been better-accepted and widely spread. There is a version of OT, however, that includes a gradual algorithm and replaces ordinal ranking with ranking along a continuous scale. See Boersma & Hayers (2001) for details. Their model, however, also differs from ours in a number of points (mainly, the conception about the initial and final number of constraints).

considered). Given this scenario, Tesar & Smolensky propose several possible algorithms (Recursive Constraint Demotion; Core Constraint Demotion; On-Line Constraint Demotion; Batch Constraint Demotion; Input/Output Constraint Demotion), but all have the following assumption in common (Tesar & Smolensky, 2000: 12):

> "***Given:***
> *universal-constraints = a set of universal constraints*
> *initial-data = a set of well-formed outputs of the target language L*
> *We assume that L arises from some (not necessarily unique) total ranking R of the universal constraints.*
>
> ***To Find:***
> *A stratified hierarchy in which these are the optimal parses of their corresponding inputs*."

Notice that the assumed part ("given") contains precisely the full amount of constraints that will be found in the "mature grammar", all that is added is the optimal hierarchy that can derive the inputs. In this sense, the OT-acquisition device as is presented by Tesar & Smolensky is more a generation procedure for hierarchies than a set of evaluation procedures for possible output grammars (see Chomsky, 1957: 51).

In Tesar & Smolensky's model there is no account of development in the sense that the number of constraints remains the same as times goes by. This, we believe, is a disadvantage insofar as the optimal scenario would be that in which all constraints are subsumed to a very local evaluating algorithm (possibly, DFI or an equivalent). A static model in which all constraints are needed at every point of acquisition is biologically implausible, even in neurologically-implemented theories of OT learning, like the one depicted in Smolensky (2006), since neural networks have memory and learning is a crucial factor when modeling such networks (see Elman, 1990; Altmann, 1997; Amari, 2003). Smolensky implements OT in a Parallel Distributed Processing neural network, distinguishing two levels: a *micro* (μ) level and a *macro* (M) level of structure processing, with only differences of scope. The link between those levels would be the notion of *harmony*, central to the OT architecture and described in the article as a "connectionist well-formedness measure" (2006: 781). Smolensky develops a formalism for linking rules between the two levels, and the algorithms that apply at both, but the problem we have outlined above is still there: the number of constraints is invariant, and their nature is just assumed. It is not clear how exactly they are to be represented in a neural network. Moreover, the system Smolensky argues in favor of (PassiveNet) produces recursive, but 2-D representations, thus limiting itself from the very definition (e.g., 2006: 789). There is no evidence to link recursiveness to bidimensionality by necessity, even if 2-D structures *may* display recursion. A theory that sees acquisition as a dynamic process of progressive constraint induction from the data and subsumption to more general and powerful constraints until getting to the optimal situation of C = 1 is not incompatible with Smolensky's, but they are complementary. In our view, the concept of learning in a PDP model is essential, as it can help simplifying the constraint system and making it psycholinguistically more plausible. A traditional model, based on *linear sequential computation*, is that outlined in Altmann (1997) and Elman (1998): neural networks encode information in the form of neural activation patterns that form a continuum, and learning (central notion in these models) is seen as an adjustment of patterns of activation, whether supervised or not: supervised learning, the system provides network input and the output expected, and the network must abstract the rule. In unsupervised learning, it is only provided input to the network, and the rule is tested a posteriori. The patterns of activation are electro-chemical stimulation must exceed a certain "threshold" to be meaningful. This threshold would be innate, the only “unlearnable” aspect in a connectionist network. Each relevant element in any mental workspace we analyze corresponds to a specific activation pattern, representing a unique electro-chemical perturbation. Networks generally consist of three layers of neurons: an input layer, a middle layer and an output layer. Suppose we have a network that interprets graphic stimuli (letters) and relates them to signifiers (acoustic images, borrowing the term from Saussure). Facing a grapheme that stimulates the activation threshold, neurons produce a given input pattern corresponding to the stimulus. This pattern is transmitted to an intermediate layer of neurons, which calculates the total activation force, and then the result is sent to an otuput layer of neurons (e.g., phonemes. See Altmann, 1997). However, the network has not learned anything yet, since activation thresholds are, in principle, arbitrary. In the mid-'80s, Jeffrey Elman proposed a connectionist model of neural networks that had memory which, exposed to linguistic stimuli, were able to "learn" to produce linear categorial expectations in the form of activation patterns (which encoded distributional information of each part of speech, clearly structural information) and to create "composite patterns" in the event that more than one possibility is feasible in a some point of the parsing process. Elman’s model incorporates to the above mechanism "copy" neurons in the intermediate layer, each connected to an intermediate neuron. The information is transmitted from input neurons to the intermediate layer, and then copied to the copy neurons. When the intermediate neurons receive new information from the input neurons also receive a copy of the prior information from copy neurons, so that the pattern of activity of neurons intermediate incorporates both the new information and the network’s past reaction to similar activation stimuli, and that is sent to the output neurons. Thus, combining patterns, the network is able to anticipate what will come in the light of the experience of what it has already parsed. As an example, let us consider the case of a root like √CAT. Is it possible to form a verbal morpheme (that is, a syntactic terminal node) to which a phonological piece /kæt/ corresponds? Yes, if the underlying construal has [CAT] (the generic concept) in a legitimate position. We cannot form a verb [$_V$ cat] from a semantic construal where CAT is on Spec-*v*, since we would be conflating the Spec into a Head, and such an operation would require many stipulations. This verb would be an *impossible word*[16]. If [CAT] is on Compl-P, for example, we could form a *locatum / location* verb [$_V$ cat] (for example, "to cat a mouse", meaning [CAUSE [GO [[mouse] [TO] [cat]]]] in Mateu Fontanals’, 2002, 2010 terms), and that would merely be a yet *uncoined word*, but perfectly *possible*, and “*parseable*”. The relation between syntactic possibilities and phonological limitations are expressed in our theory in the following form:

[16] The explanation for *impossible words* is very simple: Let us assume that we have [$_{XP}$ ZP [$_{X'}$ [X_0] YP]] and X_0 is defective, either phonologically or semantically (see Hale & Keyser, 2002). If we consider the diachronic dimension of the derivation, as soon as we have [X’ [X_0] YP], following the *Earliness Principle*, the *conflation* process must occur. There is no need (and, what is more, it would be an anti-economical option) to wait until ZP is merged. *Dynamic Full Interpretation* also plays a role, as it *triggers* conflation to generate a fully interpretable object. See below.

(26) ***Morpheme formation constraint***: *We cannot group dimensions in a terminal node (i.e., morpheme) if there is no vocabulary item in the B List specified enough to be inserted in that node.*

Corollary: *Given two sub-morphemic terminals, X and Y; relations of phonological precedence will not be determined by syntactic principles but by the availability of Vocabulary Items to Spell Out those terminals* (Cf. Embick & Noyer, 2004)

That is, DFI, combined with the MFC provides some insight onto the syntax-phonology interface, and how this constraints the possibilities of generation in a crash-proof model. Nevertheless, the *Morpheme Formation Constraint* does not help, when the morpheme has been formed according to “long-known principles of syntax” (think of GB’s principles, for example). However, *rutinized neural connections* do. Bear in mind that only Merge comes “for free” (by conceptual necessity), the "lexicon" (by which we mean the *inventory of phonological pieces*, a purely socio-historical product), is *learned*. Learning is a process of adjustment of neural connections, and when recurrent neural flows (to use a metaphor) are rutinized, the connection is made quicker, almost automatic (in a Fodorian way). Our point here is that the program of static *n*-constraints is simply biologically implausible. What we need is a system that:

(27) A. Provides means for explaining development in a dynamic way, with a variable number of constraints

B. Is fully explicitable in mathematical terms, such that it can complete a wider theory of physical systems

C. Is compatible with an FL-non-existence hypothesis (Krivochen, 2011a et. seq.), so that no central notions like “constraints” can be wiped under the UG rug.

This is the kind of model we have in mind.

Now, we will try to explicit the functioning of the system. We agree with Broekhius & Vogel (2010) in that Crash-proof does not entail filter elimination, but we will add that positing a fixed system of *n* constraints that remain the same during a *T* period of time in ontogenetic development is highly implausibly because of considerations of brain maturation: in a system with a free unbounded (and faculty-underspecified) GEN algorithm, all kinds of candidates could be assembled, but generation is constrained by legibility conditions in an *extremely local harmonic serialism* (see the sample derivation above). In a system with “invasive interfaces” like the one we are building, in which $C_{(HL)}$ has ***no specificity*** and is the result of pre-frontal neocortex activation, FL is superfluous, what is more, it is untenable under the simplest assumptions: computational effects can be achieved without the need to posit a specific and relatively autonomous Faculty of Language in Chomskyan terms. If there is no FL, then there is no UG, as UG is the theory of FL’s initial state. It is psycho-neurologically implausible that all constraints are somehow present at the initial moment of acquisition (unless artificially positing something like UG), and no compelling neurological evidence has been provided that constraints are in fact innate[17] (there is not even a consensus about how constraints would be neurologically represented). But it is also impossible that *n* stays stable during the whole time *T*, since this would mean that the neural networks have learned nothing from constant candidate filtering, and all constraints are equally necessary in every stage of development, against which we argue. For us, an optimal design would be one in which constraints are abstracted by a neural network (possibly of the kind Smolensky describes, even though we will present an alternative theory) via *non-supervised learning* in a speech community and in the end, from *n*, only *one* remains, that one which derives from the interaction between *generation* and *interpretation*: *Dynamic Full Interpretation*. Now, we will try to explicit what an RM algorithm would look like, by describing the state of the mental grammar at critical points of the development of the constraint system:

(28) Consider: $x = 1$ as the initial state of the mental system of constraints, $x = 2$ as an intermediate stage and $x = 3$ as the final mature state; furthermore, consider *n* as the maximal number of constraints.

- For $x \leq 1$, y asymptotically tends to 0
- For $x = 2$, $y = n$
- For $x \geq 3$, $y = 1$

The function described in (27) accounts for the potentiality of acquiring constraints from inductive abstraction in a speech community, as *y never equals* 0. As constraints act at the interface levels, this can be seen as the maturation of an innate, genotypic capacity in contact with some form[18] of the phenomenological world. There is a rising curve towards the maximum value in $x = 2$, an intermediate stage in the development in which all constraints are active and working. Between 2 and 3 is that stage in which so-called "redundancy rules" (or better, *procedures*) would apply: if a constraint C_1 only applies after C_2, and there is no C_3 that can also license the appearance of C_1, then C_1 is kept and C_2 is subsumed to C_1, and the system simplifies *itself*, looking for optimality. In this respect, language is *chaotic* in the technical sense: it is an *open* (i.e., permeable to external influence, like Primary Linguistic Data), *complex* (i.e., composed of subsystems, like phonology and semantics, both computational interfaces) and *dynamic* (i.e., it changes along with time) system; hypersensitive to initial conditions: those conditions that determine the ranking in the model of Tesar & Smolensky (2000) perturbate the otherwise stable initial state of the EVAL component, and the outcome is a particular ranking, corresponding to a specific natural language L. We add that acquisition cannot be complete unless, in a third stage, all constraints (some local, some global) are subsumed to a very local evaluation procedure by the interfaces, accessing the

[17] In fact, Smolensky (2006: 797, fn. 8) claims that "*To assert that a constraint is universal is to assert that it appears in the grammars of all languages—not that it is innate. OT is a theory of universal grammar in this sense: It is a theory of the universal computational properties of grammars, not a theory of innate knowledge. The source of knowledge of universal constraints is an open empirical question.*" However, it is hard for us to see how it could be an empirical question as it is unfalsifiable, or, rather, trivially verifiable: any sentence in any language can be invoked to act as a "proof" of some principle of UG, as has been extensively done.

[18] i.e., Jackendoff's "real world" or "projected world", depending on the philosophy the reader prefers, realistic or not.

generative workspace, what we have called DFI. In the derivational dynamics we have proposed, the timing of constraint-evaluation for an output is *after the application of every operation*: instead of having an "every phrase is a phase" situation, we have an "every output of Merge is a possible candidate, and thus must be evaluated". If DFI (that is, the relevant interpretative interfaces) drives the generation mechanism, the interfaces having access to the generative workspace drive the evaluation procedure –a *crash-proof* system that is guaranteed, in our model, in the final stage of the mental grammar - but what happens before that, *during* the process of language acquisition? This is the question we will address in the next section, with added considerations of the consequences the simultaneous evaluation of multiple candidates has for the model of the mind we handle.

4.1 *Turing Machines vs. the Quantum Computer*

A question is in order at this point of the discussion: we have said that OT filters out *n* candidates, which can be reduced to one should they be generated by a *crash-proof* grammar strictly following DFI. However, although we may have simplified the generation mechanisms, there is still a problem when $x \leq 2$: do constraints apply to an output serially or simultaneously? That is, given the fact that constraints are ranked, would it be possible to say that they apply *simultaneously* to an output or *in real time* during the derivational procedure? To this question, we will answer *in real time* on economy bases: step-by-step evaluation is computationally cheaper than global evaluation since there is less amount of structure active in the short time memory at any given time. Compare for example the following derivations:

(29) *[$_{PP}$ [DP_3] [[DP_2] [P [DP_1]]]] Ruled out by DFI *at the point in which DP_3 enters the derivation*. E.g.: *[[A book] [[A pen] [on [the table]]]].

(30) *[CP [TP [*v*P [VP [$_{PP}$ [DP_2] [P [DP_1]]]]]]]] Ruled out by *global evaluation procedures after the whole derivation is complete*. *v*'s requirement of an external argument, semantically interpreted as the *initiator* of the event, is not fulfilled. E.g.: *[CP [TP [*v*P [VP sent [PP [the package] [to [Mary]]]]]]]

(31) *[*v*, [V, [$_{PP}$ [DP_3] [DP_2 [P [DP_1]]]]]] Ruled out by *semi-global evaluation procedures once the vP phase is complete*, by the same reasons as (28), there is a superfluous element in the representation.

Notice that the final result is the same in all cases: the rejection of the candidate on constraint violation grounds. There is a superfluous element in the representation, namely, a nominal construction which receives no interpretation within the locative structure, since P licenses only two arguments: a *figure* (Spec-P) and a *ground* (Compl-P). Our evaluation system, while giving the same output as alternative evaluation measures, maintains less structure in the active memory, assuming (29), (30) and (31) are all "snapshots" of the same generative workspace. This, we think, is sufficient proof of the advantages of *extremely local evaluation* (although, we stress, no *optimization* is performed).

Let us compare the graphic of a typical EVAL function with our system at the "final" developmental stage $x = 3$, once all constraints have been subsumed to DFI, as in the derivation outlined in (21), following the derivational dynamics made explicit in (15):

(32)

	A	B	C
☞ O_1			*
O_2			**!
O_3		*!	
O_4	*!		
O_5		*!	*

(from Müller, 2011b: 5)

(33)

	A
☞ O_1	

In Müller's approach to OT, each candidate output O_X is evaluated by *n* constraints, which are violated and their relative hierarchical relation (A > B > C) determine fatal (!) and non-fatal (*) violations. In our system, there is no possible violation since O is evaluated by DFI at every point, thus there is no such thing as constraint violation or interface crash: this defines our system as *crash proof* in Putnam's (2010) terms. Extremely local evaluation by DFI works just fine at $x = 3$ (the situation depicted in (33)), but we have to provide a model for $x = 2$, when it is the case that $y > 1$. We have already said that global evaluation has disadvantages insofar as there is more structure active in the workspace at the evaluation point than in step-by-step evaluation, and "semi-global" evaluation has the disadvantage of requiring an *a priori* definition of transfer domains, which results stipulative in all the mainstream current proposals we are familiar with within *phase theory* (see Chomsky, 2005, 2012; Gallego, 2010; Bošković, 2012 among many others). Consequently, we will stay with the step-by-step *analyze* for our account of the constraint interaction in a RM architecture. Consider a classical linear Turing machine, which is the model Chomsky and many others have in mind when analyzing the computational properties of the human mind (see King, 1996 for a summary and a solid argumentation): how would it deal with the issue of applying several constraints to the same output representation? Two answers come to mind:

- Constraints are applied serially following the ranking, even if there is a violation in the process, until all constraints have been applied

- Constraints are applied serially following the ranking until there is a fatal violation, in which case the evaluation stops even if there are constraints left to apply

An empirical test of these claims would need a fixed set of constraints, something we do not have at present: while impressive accomplishments have been made in the field of OT-syntax (particularly, the simplification of the MP's architecture and the *generation-evaluation* dynamics, which is currently at the heart of many major models of language, like Uriagereka's 2012 CLASH model), the lack of consent with respect to the number and nature of syntactic constraints is the major obstacle for serious empirical research: nothing would prevent the selection of examples and the modeling of suitable constraints that could prove our theory, as can be seen in the literature. The mechanisms of evaluation have not been formalized either, so we do not know whether all candidates are evaluated at once or sequentially, and what motivates the sequence if the second scenario is correct. If all candidates are part of the same reference set, we have no reason to establish an order, unless we explicitly pick one and apply all constraints to it. In doing that, however, we are making a stipulation, but not answering the question on principled grounds. We will thus remain at the level of theoretical modeling, which has the advantage of allowing evaluation on independent grounds, like logical consistence and theoretical economy. Given this two scenarios, *linear sequential evaluation* and *simultaneous multiple evaluation*, a Turing program for linguistic theory (TPLT) has been attempted (see Watumull, 2012a and references therein) –corresponding to the first scenario-, and we will propose an equally summary and programmatic *quantum* alternative here, following the claims in Krivochen (2011a, b, 2012a, b) –corresponding to the second-.

Watumull's *TPLT* is based upon the assumption that cognition is computational, and thus definable as a set of algorithms. We share those assumptions, as has been made explicit in Krivochen (2011b), but differ from Watumull (and most current research in the computational basis of language) in that we take computations to be quantum-like rather than linear, digital-like. A Quantum Human Computer (QHC) does not base itself on bottom-up derivation by binary Merge, but works with *parallel workspaces* deriving by *n*-ary concatenation (*n* being determined by interface conditions on legibility[19]) of objects which, *in abstracto*, comprise all possible

[19] Watumull (2012b: 314) points out that "*For n-ary Merge, the set of possible values of n—equivalent to the set of possible procedures for computing an output—is infinite and the input k can be infinite (e.g., were SOs to be continually selected and merged from the lexicon and/or parallel derivations). If n = k, k = ∞, then n-ary Merge cannot be finitely defined—assuming as I am, in this instance, that the function is not decomposed into a sequence of finitely defined operations—and thus the procedure cannot halt to produce an output.*" There are several objections to this reasoning. The first one is the "if-" clause that equates *n*, *k* and ∞, given the fact that it is impossible, either in orthodox Minimalism or Radical Minimalism, that the system works with infinite units: units are always drawn from the lexicon (which is finite) and they are instantiated as *tokens* following interface conditions (i.e., how many tokens of each type are required to conserve the information throughout the derivational path), which are also finite. Therefore, there is no way in which *k* could be infinite in language. The only mental capacity in which $k = \infty$ is in the arithmetical capacity, and we have tackled the issue in Krivochen (2012a) when considering Tegmark's (2007) problems with the Mathematical Universe Hypothesis. Notice that a halting algorithm is only required when there is a generation-interpretation interface, otherwise, there is no non-stipulative way to stop generation (and no requirement to do so). Consider a simple linear function $f(x) = 2x$: is there any halting algorithm for it? If we graph the function, limitations are given by, for instance, the size of the paper, crucially not by any system-internal algorithm. There is no halting function because it is not needed.

outcomes. The classic example is Case in nominals: any nominal construction can appear in a particular derivation and it will be interpreted as Nominative (Initiator), Accusative (Theme, Instrument) or Dative (Location, Possession). However, it would be inaccurate to say that the construction has no Case *in abstracto*, since if it had no Case at all, it could not be "given" Case in the derivation as it would violate the ConsP (or Chomsky's Inclusiveness Condition), information *cannot be created*, only transformed. On the other hand, the [u-T] Pesetsly & Torrego (2007) propose requires an Agree mechanism, with all concomitant substantive complications (the presence of certain features in both a "probe" and a "goal", feature copying / sharing / donating, values, interpretability issues, transfer timing, etc.) besides the methodological disadvantage of restricting the GEN mechanism stipulatively (as pointed out by Boeckx, 2010b). Our reasoning, which has been presented in Krivochen (2011a), is as follows: imagine we have an electron in a tridimensional space, and we want to know its location. In order to do so, we need to see it, projecting some kind of light on it. This light is projected in the form of a *photon*, a particle with mass. The "problem" is that when the photon crashes with the electron, there is a change in the original location, which remains unknown. That original location (we have taken this magnitude just for the sake of the example, but we could have also worked with speed or trajectory) is taken to be a "superposition" of all possible locations, expressed in the form of a "*wave function*" (in de Broglie's terms). Therefore, there will always be a magnitude whose real value will remain unknown to us. In this kind of physical systems, it is the ***observation*** that makes the relevant dimension *collapse* to one of the possible states[20]. *Uncertainty* (as defined by Heisenberg, 1999) *is a natural characteristic of physical systems, and by no means an instrumental problem*, taking physical system in its technical sense, that is, any portion of the physical universe chosen for analysis. We take "physical universe" to be equivalent to "natural world", and we will use one or the other indistinctly. Magnitudes (or *dimensions*, to maintain a term more closely related to linguistics, since we are not dealing with measurable elements) are *not* necessarily binary; what is more, *in abstracto* they can comprise as many states as the system requires, which, as we will show later, leads us to a much simpler form of minimalism. ***We will express it by using this notation: for any dimension D, [D_X] expresses its quantum state.***

Let us suppose that we have a physical system which starts out in a state α, and changes, over some time, into state α'. Of course, it could have started out in any of many *different* states. So suppose it starts out in state β, and changes over the same considered time interval into state β'. We can schematically represent these two possible "trajectories" like this:

(34) $\alpha \rightarrow \alpha'$

(35) $\beta \rightarrow \beta'$

[20] See, for example, the well-known EPR (Einstein-Podolsky-Rosen) paradox, which inspired Schrödinger (1935) paper.

Since α and β are possible states of the system, so is their arbitrary linear combination **aα + bβ**. What ***Schrödinger's Equation (SE)*** tells us is that given that α and β would change in the ways just indicated, their linear combination must also change in the following way:

(36) $\mathbf{a}\alpha + \mathbf{b}\beta \longrightarrow \mathbf{a}\alpha' + \mathbf{b}\beta'.$

The interesting fact about the above mentioned equations is that *they hold only if no "measurement" is taking place.*

If a "measurement" (say, interpretation by an interface system) is taking place then we must consider an entirely different story about how the state of the system changes: during the measurement, the system S must "*collapse*" into a state that is certain to produce the observed result of the measurement. The hypothesis is exemplified by Schrödinger (1935) using the now famous "cat paradox", which deserves to be quoted in full-length:

> *"A cat is penned up in a steel chamber, along with the following device (which must be secured against direct interference by the cat): in a Geiger counter there is a tiny bit of radioactive substance, so small, that perhaps in the course of the hour one of the atoms decays, but also, with equal probability, perhaps none; if it happens, the counter tube discharges and through a relay releases a hammer which shatters a small flask of hydrocyanic acid. If one has left this entire system to itself for an hour, one would say that the cat still lives if meanwhile no atom has decayed. The ψ-function of the entire system would express this by having in it the living and dead cat (pardon the expression) mixed or smeared out in equal parts. It is typical of these cases that an uncertainty originally restricted to the atomic domain becomes transformed into macroscopic uncertainty, which can then be* ***resolved*** *by direct observation*". (p. 7-8. Highlighted in the original)

The question to be asked now is: *how do we apply this to language?*

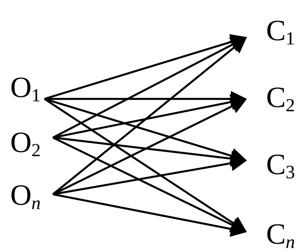
Our answer will be the following: ***we will consider language to be a physical system, and therefore, if SE applies to any physical system, it must also apply to language***. Thus, our conception of abstract Case in nominals, as well as other phenomena that so far were assumed to involve features: categorization, Minimality effects, binding theory, displacement in general (see Krivochen, 2011b, 2012b for details). A Turing machine, as far as we understand, is incompatible with this kind of non-linear computation, in which a single algorithm computes elements comprising states which collapse only in a particular derivation and in a local relation with specific procedural nodes. We have analyzed the phenomenon we have called "influence" elsewhere, we will now shift our focus to how a quantum-like EVAL would work when *all constraints are active*. Let us imagine the following situation: we have *n* candidates and *x* constraints, where both *n* and *x* > 1. For all natural languages NL and all constraints *x*, we assume that:

(37) $\forall(x)\ \&\ \forall(\mathrm{SO})\ |\ \mathrm{SO} \in \mathrm{NL},\ x(\mathrm{SO})$

(37) is a crucial assumption for OT syntax, since it guarantees universality as well as the application of the constraints to all syntactic objects (SO) belonging to a certain NL (that is why we used the predicate-argument notation). We now face the following problem: how and when do those constraints apply? One possibility is the Turing machine procedure, the serial and linear application of the EVAL algorithm, compatible with (26) – (28). However simple this option may seem, we find it difficult to implement it in the kind of neural networks we talked about in section 4: even if statistical learning can be codified in a Turing machine, learning timing seems to pose a difficult counterargument for the linear theory. We could certainly have several parallel processors performing Turing-like computations, but the theory would be complicated as new questions would arise: are all constraints present in every processor? How are candidates divided between the processors? Do we need to determine a reference set for each processor? The EVAL function, which is assumed to have neural basis, is determined by the limitations the hardware imposes the software (i.e., the characteristics of the neural networks we assume will impact directly on the computational possibilities they license). Another, quite different possibility is that, at each derivational point, evaluation takes place *simultaneously* for *several candidates* in a *single workspace*.

Consider the determining of *transderivational* constraints (see section 2.2): how come multiple derivations be compared with respect to some aspect A, which should be a constraint, in OT terms, and a winner is selected among them without it taking literally an eternity (bear in mind we have infinite use of finite media, the number of possible derivations of a reference set can always be infinite)? The reference set is unary only when x = 3, that is, when the mental grammar is indeed *crash-proof* and the only constraint is DFI. Acquisition, as we have already pointed out, can be seen in this framework as the process via which the constraints that apply to (partial) representations are subsumed to a single, general condition on structure building and interpretation, DFI: during this process, which has been neurologically analyzed as threshold adjustment for neural excitation, combining stimulation (through different levels) with inhibition (among units of the same level, once one paradigmatic option is chosen), multiple constraints must be considered, having been inducted from primary linguistic data. This situation can be represented in the following form:

(38)



The problem is, how do we make our network process multiple candidates *simultaneously*? Assume:

O_1 = What did you buy from the store?

O_2 = You did buy what from the store?

O_3 = What you bought from the store?

And, considering some of the constraints we have been exemplifying from the literature (mainly, Kager, 1999 and Müller, 2011b), the EVAL component would be:

C_1 = Wh-CRIT (Wh- appear in Spec-C)

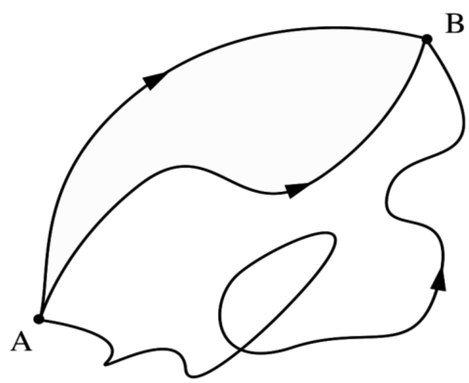
C_2 = θ-ASSIGN (Elements appear in the site in which they receive a theta-role)

C_3 = Full Interpretation (no superfluous elements in representations: traces, expletives, etc.)

C_4 = Subjacency[21] (movement cannot cross more than one bounding node)

This state of affairs is a very limited representation, but it will do for our purposes. The complex network of interactions between GEN and EVAL depicted in (38) should be performed in real time, so, how does the system select O_1 among its competitors? Our proposal, to be confronted with linear Turing-machine processors, is to think of this situation in the same terms in which Feynman thought about particle movement: the so-called *path integral formulation* (e.g., Feynman, 1948; Rattazzi, 2009). The rationale is as follows: instead of *measurement* in charge of determining the specific location of (say) a photon in T, being its previous location a *wave function*; what we have is a functional integral over an infinity of possible trajectories to compute a quantum amplitude. The formulations are equivalent within quantum mechanics, and they both follow from Schrödinger's Equation, thus being incompatible with traditional particle models. A common representation of Feynman's approach is the following:

(39)



[21] Depending on the reader's preference, Subjacency could also be expressed in terms of the Phase Impenetrability Condition or Relativized Minimality, among others. RM has the advantage of not defining *a priori* the identity of the intervenient nodes, whereas in Subjacency and PIC, the nodes are pre-defined (S and NP in the case of Subjacency and *v* and C in the case of the PIC).

Of course, the indicated paths are just three of infinite possible paths. The question now is how to implement this in a model of language, in which "infinity" is a notion strictly restricted to the outputs of a generative algorithm, slightly modifying Humboldt's approach. Mental entities are ultimately electricity, obtained through neural stimulation/inhibition as a function of the presented input: as they are ultimately *energy*, they cannot be created nor destroyed, just transformed (what we expressed via ConsP); and in a neural network the GEN-EVAL dynamics are to be represented as the translation from A to B in (39), being A and B points in a specific network within the phase space of all possible connections, of which the contact with the phenomenological world reinforce only one, which we can call an "optimal network" given input and output conditions[22]. Being A the set of candidates and B the EVAL component, each candidate is *simultaneously* applied all constraints by EVAL, based on the rationale behind the path integral formulation, and *all candidates* undergo this process simultaneously, O_1 being determined to be the optimal candidate much quicker than in the alternative scenario. Such computational method is consistent with the formulations of the *Quantum Human Computer Hypothesis* and other proposals about quantum computation (e.g., Rasmussen et. al., 1990, particularly focused on neural networks), whereas it is dubious whether it can be applied, maintaining the theoretical advantages and empirical prospects, in a Turing-machine model of language like that argued for in Watumull (2012a). However, the possibility cannot be dismissed. Comparison and discussion will be left for further investigation.

A note is to be made regarding data: the present paper is a theoretical analysis of the foundations of OT, and a discussion of how it could be reconciled with models of other physical systems, if language is indeed taken to be part of the natural world: our inquiry has been triggered by the fact that so-called "explanation" in Linguistics differs in unbelievable ways from explanation in the natural and formal sciences, thus making it difficult to integrate the study of language in the study of biological systems (in a very narrow way) in some other way than the metaphor, which is, to us, of little use. For when it comes to find the biological correlates of [u-F], for example, there is no single attempt that fulfills basic scientific desiderata, like logical consistency, decidability, economy[23]. Being ours an inquiry on the foundations of OT, and a discussion of its basic assumptions from a biological-physical perspective (in permanent dialogue with computational science[24]), there is no point in discussing data as it has been already done with varying degrees of success: for the purposes of our paper, data analysis is a future step but "empirical predictions" do not change with respect to traditional OT. We have taken a formal system and simplified it, as well as enriched it with interdisciplinary discussion, maintaining descriptive and explanatory adequacy: by all means, a valid scientific move (Lakatos, 1978a, b).

[22] In this sense, we are assuming a generalized-delta rule in a supervised learning model. Nevertheless, the model can be applied in a non-supervised learning model without major adjustments.
[23] The recent introduction of "Münchhausen features" (Fanselow, 2002) and "Ghost PP deletion" (Collins & Radford, 2012) can serve as examples of unnecessary complication of the jargon with little or no impact in the explanatory power of the theory.
[24] We have compared a Turing program with a Quantum program as we consider them to be the most interesting novel approaches to computational models of the mind. For a more traditional computational approach based on a digital computer, see Fitchner (2005).

5. Conclusion: Radically Minimalist OT as a model for crash-proof grammars

The present paper has tried to make a point of the fact that the human mind is a computational system which works like a quantum computer. This conception of the functioning of the human mind has been exemplified with an OT architecture, in which the nature and form of constraints determine the interaction between generative and interpretative systems. What is more, the QHC can be the theoretical background upon which to model a theory of learning with statistical basis, in which the progression from the initial, genotypic state to the final, relatively stable state of the system is a function describing neural reconfiguration in a chaotic system. Once all other constraints are subsumed to DFI, we have a system with the advantages of *both* free generation *and* crash-proof systems (Cf. Boeckx, 2010a):

- GEN is mathematically defined as a universal, unbounded and free concatenation function.
- EVAL is dynamically conceived, which gives OT more biological plausibility: P&P's static character is replaced by a dynamic system of constraints, sensitive to external factors during a time T. What is more, we have introduced a recent debate between digital, Turing-like computers and non-linear computers and the consequences the adoption of one and the other would have for the modified OT architecture we have outlined.
- DFI is a third-factor constraint, since it depends on *interface effects*: if the application of an operation generating SO_1 does not increase the informational load of SO_{-1}, it is an *illegitimate* step, and thus should not apply at all. This allows us to permit *soft crash* effects while still having a crash-proof syntax in the relevant sense, within *local* domains. This is an improvement over current versions of *harmonic serialism*, in which evaluation and optimization tends to be *too local* to be non-trivial or just crash-rife.
- Our feature free model is theoretically simpler than feature rich alternatives like Heck & Müller's (2007) and Müller (2011a, b). Empirical evidence can be accounted for by both models (needless to say, the creation of *ad hoc* features can enhance descriptive adequacy and generate the illusion of explanatory adequacy, even if all that is actually done is to take some data and create a set of theoretical elements that suit that data instead of proceeding in a hypothetical-deductive manner), and at this point in the development of models for natural language acquisition we believe it is essential to present alternatives which question the foundations of competing theories and problematize that which is usually taken for granted.

The potential advantages of adopting a Radically Minimalist version of OT are clear, we think. The result is a non-stipulative crash-proof model of syntax, including a model of acquisition, which is fully explicit in mathematical terms, via which we are hopefully a step closer to a higher level of scientific abstraction, beyond *explanation*: *justificative adequacy*, studying language as a physical system, instantiating a mathematical structure. Our only hope is that this is the first of many interdisciplinary works in the topic, and bridges between Linguistics, Physics and Mathematics can be built beyond specific self-centered theories and stipulations.

6. Bibliography:

Amari, S. (2003) Learning and Statistical Inference. In Arbib (2003). 624-628.

Arbib, M. (2003) *The Handbook of Brain Theory and Neural Networks*. MIT Press.

Altmann, J. (1997) *The Ascent of Babel: An exploration of language, mind, and understanding*. Oxford: Oxford University Press.

Barbosa, P. et al. (eds.) (1998). *Is the best good enough? Optimality and competition in syntax*. Cambridge, Mass.: MIT Press.

Bernardez, E. (1982) *Introducción a la Lingüística del Texto*. Madrid.

(2001) *De monoide a especie biológica: aventuras y desventuras del concepto de lengua*. Ms. Universidad Complutense de Madrid.

Boeckx, C. (2007) “Eliminating Spell-Out”. *Linguistic Analysis* 33 (3‑4) (Dynamic Interfaces, K.K. Grohmann (ed.)): 414‑425

(2010a) “A tale of two minimalisms: Reflections on the plausibility of crash-proof syntax, and its free-merge alternative”. In Putnam, M. (ed.), 105-124.

(2010b) Defeating Lexicocentrism. Ms. http://ling.auf.net/lingbuzz/001130

Boersma, P., J. Dekkers & J. van de Weijer (2000) “Introduction”. In *Optimality Theory: Syntax, Phonology and Acquisition*. Oxford, OUP.

Boersma, P. & B. Hayes (2001) Empirical tests of the Gradual Learning Algorithm. *Linguistic Inquiry* 32: 45–86.

Bošković, Ž. (2012) “Now I’m a phase, now I’m not”. Ms. http://ling.auf.net/lingbuzz/001437. To appear in Linguistic Inquiry.

Broekhuis, H (2006) “Derivations (MP) and Evaluations (OT)”. In *Optimality Theory and Minimalism: A possible Convergence?* Linguistics in Potsdam 25. 137-193.

Broekhuis, H & R. Vogel (2010) “Crash-proof syntax and filters”. In Putnam, M. (ed.), 245-267.

Carnap, R. (1966) *Philosophical Foundations of Physics*. Martin Gardner, ed. Basic Books.

Chomsky, N. (1957) *Syntactic Structures*. Berlin. Mouton de Gutyer.

(1977) "On Wh-Movement". In: P. Culicover, T.Wasow & A. Akmajian, eds., *Formal Syntax*. Academic Press, New York, pp. 71–132.

(1995) *The Minimalist Program*. MIT Press.

(2005) "Three Factors in Language Design". *Linguistic Inquiry* 36, 1–22.

Collins, C. & A. Radford (2012) Gaps, Ghosts and Gapless Relatives in Spoken English. http://ling.auf.net/lingbuzz/001699

Di Sciullo, A-M. & D. Isac, (2008) "The Asymmetry of Merge". In *Biolinguistics* 2.4: 260–290

Embick, D. (2010) *Localism versus Globalism in Morphology and Phonology*. Cambridge, Mass. MIT Press.

Epstein, S. & T.D. Seely (2006) *Derivations in minimalism.* Cambridge: CUP.

Fanselow, G. (2002). Münchhausen style head movement and the analysis of verb second. In: Anoop Mahajan, (ed.), *Proceedings of the Workshop on Head Movement*, 40-76. Los Angeles: UCLA, Linguistics Department.

Feynman, R. P. (1948). The Space-Time Formulation of Nonrelativistic Quantum Mechanics. *Reviews of Modern Physics* 20 (2): 367–387.

Fitchner, J. (2005) *Optimality Theory and Computational Linguistics: An Overview*. MA Thesis, University of Zurich. http://www.cl.uzh.ch/studies/theses/lic-master-theses/lizjohannfichtner.pdf

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

Grimshaw, J (1997) "Projection, Heads, and Optimality", *Linguistic Inquiry* 28, 373–422.

(1998) *Constraints on Constraints in Optimality Theoretic Syntax*. Ms., Rutgers University,

New Brunswick, New Jersey

Greene, B. (1999) *The Elegant Universe*. New York: W. W. Norton.

Hale, K. & J. Keyser (2005) "Aspect and the Syntax of Argument Structure". In *The Syntax of Aspect. Deriving Thematic and Aspectual Interpretation*, N. Erteschik-Shir & T. Rapoport (eds). Oxford/New York: Oxford University Press. 11-41.

Heck, F & G Müller (2007) “Extremely Local Optimization”. In: E. Brainbridge & B. Agbayani, eds., *Proceedings of the 26th WECOL*. California State University, Fresno, pp. 170–183.

(2010) “Extremely Local Optimization”. Ms., Universität Leipzig. To appear in Hans Broekhuis and Ralf Vogel (eds.), *Derivations and Filtering*. Equinox Publishing.

Hoeks, P. & P. Hendriks (2011) Optimality Theory and Human Sentence Processing: Towards a Cross-Modular Analysis of Coordination. In *Journal of Cognitive Science* 12 (1): 83-128.

Kager, R. (1999) *Optimality Theory*. Cambridge University Press, Cambridge.

Jakubowicz, C. (2011) Measuring derivational complexity: New evidence from typically developing and SLI learners of L1 French. *Lingua*, 121. 339-351.

King, D. (1996) Is the Mind a Turing Machine? In *Synthese* Vol. 108, No. 3, Computation, Cognition and AI (Sep., 1996), pp. 379-389.

Krivochen, D. (2011a) An Introduction to Radical Minimalism I. *IBERIA*, n°3. Vol 2.

(2011b) The Quantum Human Computer Hypothesis and Radical Minimalism: A Brief Introduction to Quantum Linguistics. In *International Journal of Linguistic Studies* Vol. 5 n° 4, 87-108.

(2011c) The syntax of Spanish relational adjectives. *Sorda & Sonora* 2(1): 15-28

(2012a) Towards a Geometrical Syntax. Ms. Universität Potsdam. Submitted.

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

Krivochen, D. & P. Kosta (to appear) *Eliminating Empty Categories.* Potsdam Linguistic Investigations 10. Peter Lang.

Lakatos, I. (1978a) *The Methodology of Scientific Research Programmes: Philosophical Papers Volume 1*. Cambridge: Cambridge University Press

(1978b) *Mathematics, Science and Epistemology: Philosophical Papers Volume 2*. Cambridge: Cambridge University Press.

Leung, T. (2010) On the mathematical foundations of crash-proof grammars. In Putnam, M. (ed.), 213-244.

Mallot, H. (2003) Layered Computation in Neural Networks. In Arbib (2003). 635-638

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

(2010) On the L-Syntax of Manner and Causation. In M. Duguine, S. Huidobro & N. Madariaga (eds.). *Argument structure and syntactic relations.* Amsterdam / Philadelphia: John Benjamins. 89-112.

McCarthy, J. (2009) Harmonic Serialism in Optimality Theory. Presented at the Second Theoretical Phonology Conference, NCCU. January 10th, 2009.

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

(2011b) *Optimality Theoretic Syntax*. Ms. Universität Leipzig.

Ott, D. (2012) *Local instability: Split topicalization and quantifier float in German*. Berlin/New York: de Gruyter

Panagiotidis, P. (2010) *Functional Heads, Agree and Labels*. Ms. University of Cyprus.

Pesetsky, D. & E. Torrego (2007) "The syntax of valuation and the interpretability of features". In: S. Karimi, V. Samiian & W. K. Wilkins, eds., *Phrasal and clausal architecture: Syntactic derivation and interpretation*. John Benjamins, Amsterdam, pp. 262–294.

Prince, Alan & Paul Smolensky (2004): *Optimality Theory. Constraint Interaction in Generative Grammar*.

Blackwell, Oxford.

Prusinkiewicz, Przemyslaw; Lindenmayer, Aristid (1990). *The Algorithmic Beauty of Plants*. Springer-Verlag. [Electronic edition: 2006]

Putnam, M (ed.) (2010) *Exploring Crash Proof Grammars*. Amsterdam, John Benjamins. [Language Faculty and Beyond, 3].

Putnam, M. (2010) "Exploring crash-proof grammars: An introduction". In Putnam, M. (ed.), 1-14.

Putnam, M. & T. Stroik (2010) "Syntactic relations in Survive-minimalism". In Putnam, M. (ed.), 143-166.

(2011) *Syntax at Ground Zero*. Ms. Penn State University & University of Missouri-Kansas. To appear in *Linguistic Analysis*.

Rasmussen, S., H. Karampurwala, R. Vaidshanath, K. Jensen, and Hameroff, S. (1990) "Computational Connectionism within Neurons". *Physica D: Nonlinear Phenomena* 42 (1–3): 428–449

Rattazzi, R. (2009) *The Path Integral approach to Quantum Mechanics*. Lecture Notes for Quantum Mechanics IV, Institute of Theoretical Physics, Laussanne.

Schrödinger, E. (1935) The Present Situation in Quantum Mechanics. In *Proceedings of the American Philosophical Society*, 124.

Shannon, C. (1948) *A Mathematical Theory of Communication*. Bell System, Technical Journal.

Smolensky, P. (2006) "Harmony in Linguistic Cognition", *Cognitive Science*, 30 (5), 779 – 801.

Smolensky, P., & Legendre, G. (2006a). *The harmonic mind: From neural computation to optimality-theoretic grammar: Vol. 1. Cognitive architecture. Vol 2: Linguistic and philosophical implications.* Cambridge, MA: MIT Press.

Talmy, L. (2000) *Toward a Cognitive Semantics*. Cambridge, MA: MIT Press.

Thornton, R., & Tesan, G. (2007) "Categorical acquisition: Parameter setting in Universal Grammar". *Biolinguistics*, 1.1, 49-98.

Tesar, B. & P. Smolensky (2000) The Learnability of Optimality Theory: An Algorithm and Some Basic Complexity Results. Rutgers Optimality Archive: http://roa.rutgers.edu/files/2-0000/2-0000-TESAR-0-0.PDF

Uriagereka, J. (2002) Multiple Spell-Out. In Uriagereka, J. *Derivations: Exploring the Dynamics of Syntax*. London, Routledge. 45-65.

(2007) Clarifying the Notion "Parameter". *Biolinguistics*, 1.1, 99-113.

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

Vogel, R (2006) The Simple Generator. In Hans Broekhuis and Ralf Vogel (eds.): *Optimality Theory and Minimalism: A Possible Convergence?*. LiP 25. Potsdam.

Wartena, C. (2000) A note on the generative capacity of optimality systems. In Blutner & Jaeger (Eds.) *Studies in Optimality Theory*. LiP 8. Potsdam.

Watumull, J. (2012a) A Turing Program for Linguistic Theory. In *Biolinguistics*, 6.2. 222-245.

(2012b) The Computability and Computational Complexity of Generativity. COPiL 11. 311-329.

Wilson, D. & D. Sperber (2004) Relevance Theory. In Laurence Horn y Gregory Ward eds., *Th e Handbook of Pragmatics*, Blackwell, Oxford, 2004, pp. 607-32.