OLD AND NEW MINIMALISM: A HOPF ALGEBRA COMPARISON

MATILDE MARCOLLI, ROBERT C. BERWICK, NOAM CHOMSKY

Abstract. In this paper we compare some old formulations of Minimalism, in particular Stabler's computational minimalism, and Chomsky's new formulation of Merge and Minimalism, from the point of view of their mathematical description in terms of Hopf algebras. We show that the newer formulation has a clear advantage purely in terms of the underlying mathematical structure. More precisely, in the case of Stabler's computational minimalism, External Merge can be described in terms of a partially defined operated algebra with binary operation, while Internal Merge determines a system of right-ideal coideals of the Loday-Ronco Hopf algebra and corresponding right-module coalgebra quotients. This mathematical structure shows that Internal and External Merge have significantly different roles in the old formulations of Minimalism, and they are more difficult to reconcile as facets of a single algebraic operation, as desirable linguistically. On the other hand, we show that the newer formulation of Minimalism naturally carries a Hopf algebra structure where Internal and External Merge directly arise from the same operation. We also compare, at the level of algebraic properties, the externalization model of the new Minimalism with proposals for assignments of planar embeddings based on heads of trees.

1. Introduction

The *Minimalist Program* of generative linguistics, introduced by Chomsky in the '90s, [8], underwent a significant simplifying reformulation in the more recent work [9], [10], [11], [12] (see also [4], [5], [31]). A recent overview of the current formulation of the fundamental Merge operation of the Minimalist Model is presented in the lectures [13].

In our paper [37] we showed that the new formulation of Merge has a very natural mathematical description in terms of magmas and Hopf algebras. This mathematical formulation makes it possible to derive several desirable linguistic properties of Merge directly as consequences of the mathematical setting. We also showed in [37] that the mathematical formulation of Merge has the same structure as the mathematical theory underlying fundamental interactions in physics, such as the renormalization process of quantum field theory and the recursive solution of equations of motion via combinatorial Dyson-Schwinger equations. An analogous recursive generative process of hierarchies of graphs plays a crucial role in both cases, as Feynman graphs in the physical case and as syntactic trees in the linguistic case.

In the present companion paper, we show how this same mathematical formalism based on Hopf algebras can be used to compare the new formulation of Merge with older forms of the Minimalist Model. The advantages of the new formulation can be stated directly in linguistic terms, as discussed in [10], [11], [13], for instance. What we argue here is that one can also see the advantage of the New Minimalism directly in terms of the underlying mathematical structure.

More precisely, we first consider the case of Stabler's "Computational Minimalism", [42]. This formulation is unsatisfactory from the linguistic perspective, and superseded by Chomsky's more recent formulation of Minimalism. We take this as an example because among the older versions of Minimalism it is one that tends to be more widely known to mathematicians (as well as to theoretical computer scientists), through its relation to formal languages. Indeed, mathematicians familiar with the theory of formal languages are usually aware of the fact that Stabler's formulation

Date: 2023.

of Minimalism is describable in terms of a class of *minimalist grammars* (MG), which are equivalent to *multiple context free grammars*, a class of context-sensitive formal languages that include all the context-free and regular languages, as well as other classes such as the tree-adjoining grammars, see [44]. The MG grammars can also be characterized in terms of linear context free rewrite systems (LCFRS), [39]. However, as shown by Berwick in [3], this equivalence between Stabler's computational minimalism and multiple context free grammars hides an important difference in terms of "succintness gap". Namely, computational minimalism is exponentially more succinct than otherwise equivalent multiple context free grammars, (see [3] and [42]). As shown in [3], a similar gap exists between transformational generative grammar and generalized phrase structure grammar. Another problem with thinking in terms of formal languages is that they are designed to describe languages as strings (oriented sets) produced as ordered sequences of transitions in an automaton that computes the language. This time-ordered description of languages hides its more intrinsic and fundamental description in terms of structures (binary rooted trees without an assigned planar embedding, in mathematical terms). Indeed the current form of Minimalism not only to provides a more efficient encoding of the generative process of syntax, but it also proposes a model where the core computational structure of syntax is entirely based on structures rather than on linear order. The latter (equivalently, planar embeddings of trees) is superimposed to this core computational structure, in a later externalization phase. Thus, the MG grammars description of Stabler's minimalism, comfortingly familiar as it may appear to the mathematically minded, is in fact deceiving, since formal languages only deal with properties of terminal strings, not with structure, while Minimalism is primarily concerned with structure and does not need to incorporate linear order in its computational core mechanism.

The goal of the present paper is to elaborate on this difference. We also want to stress the point that, while formal languages have long been considered the mathematical theory of choice for application to generative linguistics, it is in fact neither the only one nor the most appropriate. As we argued in [37], the algebraic formalism of Hopf algebras is a more suitable mathematical tool for theoretical linguistics, and a useful way to move beyond the traditional thinking in terms of formal languages. Thus, in the present paper, we will use this Hopf algebras viewpoint to make comparisons between older forms of minimalism like Stabler's and the current form based on free symmetric Merge, and to show the advantages of the latter from this algebraic perspective.

We analyze Stabler's formulation of Minimalism from the point of view of Hopf algebra structures, which we have shown in [37] to be an appropriate algebraic language for the formulation of Merge in Minimalism. We show here that, in the older setting of Stabler's Computational Minimalism, Internal and External Merge correspond to very different types of mathematical structures. External Merge is expressible in terms of the notion of "operated algebra", while Internal Merge can be described in terms of right-ideal coideals in a Hopf algebra, and corresponding quotient right-module coalgebras. While these are interesting mathematical structures, the very different form of Internal and External Merge makes it difficult to reconcile them as two forms of a single underlying basic Merge structure. This is unsatisfactory, because linguistically one expects Internal and External merge to be manifestations of the same fundamental computational principle.

We then show that the mathematical formulation of the New Minimalism that we introduced in [37] completely bypasses this problem, by directly presenting a unified framework for both Internal and External Merge.

We also further discuss our proposed model for externalization of [37], by comparing it with proposed alternative models based on possible constructions of planarizations of trees, independent of syntactic parameters. We outline some difficulties with such constructions, at the level of the underlying algebraic structure.

2. Stabler's Minimalism and the Loday–Ronco Hopf algebra

In this section we first recall some mathematical structures, in particular the Loday–Ronco Hopf algebra of *planar* binary rooted trees, and the explicit form of product and coproduct. In §2.1 we recall the basic definition and properties of Hopf algebras and in §2.2 we introduce the specific case of binary planar rooted trees, with the product and coproduct described in §2.4.

It is important to point out here that one way in which older formulation of Minimalism differ from the more recent formulation is in the fact of considering *planar* trees, namely rooted trees endowed with a particular choice of a *planar embedding*. Fixing the embedding is equivalent to fixing an ordering of the leaves of the tree, which corresponds linguistically to considering a linear order on sentences. While this is assumed in older versions of Minimalism, the newer version eliminates the assignment of planar structures at the level of the fundamental computational mechanism of Merge, relegating the linear ordering to a later externalization procedure that interfaces the fundamental computational mechanism of Merge (where ordering is not assumed) with the sensory-motor system, where ordering is imposed in the externalization of language into speech, sign, writing. The difference between imposing a priori a choice of planar embeddings on trees or not leads to significantly different mathematical formulations of the resulting Hopf algebras and Merge operations.

Another very significant difference between older versions of Minimalism and the newer formulation is the calculus of labels of syntactic features associated to trees, and how such labels determine and are determined by the transformation implemented by Internal and External Merge. We introduce labeling in §2.5. For a discussion of *labels and projection* in the different versions of Minimalism see [9].

We show in §2.6 how in older formulations of Minimalism such as Stabler's, the conditions on planarity and labels impose conditions on the applicability of Merge operations, for both External and Internal Merge, and that this makes the operations only partially defined on specific domains, reflecting conditions on matching/related labels and on projection. We discuss the specific case of External Merge in §2.7 and of Internal Merge in §2.8.

In §2.9 and §2.10 we describe the mathematical structure underlying the formulation of Internal Merge in the older versions of Minimalism. In §2.9 we show that the issue of labels and domains requires a modification of the Loday–Ronco coproduct, while in §2.10 we show that a similar modification is required for the product, and that the problem of heads and projection results in the fact that the domain of Internal Merge is a right ideal but not a left ideal with respect to the product, which is a coalgebra with resoect to the modified coproduct.

Moreover, in §2.11 we show that the presence of domains that make Merge partially defined, due to labeling conditions, creates an additional mathematical complication related to the iterated application of Merge, as composition of partially defined operations.

We observe in §2.12 that a significant difference with respect to the Hopf algebra formulation of the new Minimalism we described in [37] arises in comparison with the analogous structure in fundamental physics. In the case of the old Minimalism discussed here the Internal Merge has an intrinsically asymmetric structure, because of constraints coming from labeling and projection and from the imposition of working with trees with planar structures. Because of this, instead of finding Hopf ideals as in the setting of theoretical physics, one can only obtain right-ideal coideals, which correspond to a much weaker form of "generalized quotients" in Hopf algebra theory.

We then discuss in more detail the mathematical structure of the old formulation of External Merge in §2.13 and §2.14. We show that these exhibit a very different mathematical structure with respect to Internal Merge. It can be described in terms of the notion of "operated algebra", where again one needs to extend the notion to a partially defined version because of the conditions on domains coming from the labels and projection problem.

2.1. **Preliminaries on Hopf algebras of rooted trees.** We begin by reviewing the main constructions of Hopf algebras of rooted trees, focusing on the *Loday–Ronco Hopf algebra*, which is based on *planar binary rooted trees*.

Associative algebras can be regarded as the natural algebraic structure that describes linear strings of letters in a given alphabet with the operation of concatenation of words. On the other hand, in Linguistics the emphasis is put preferably on the syntactic trees that provide the parsing of sentences, rather than on the strings of words produced by a given grammar. When dealing with trees instead of strings as the main objects, the appropriate algebraic formalism is no longer associative algebras, but Hopf algebras and more generally operads. These are mathematical structures essentially designed for the parsing of hierarchical compositional rules, by encoding (in a coproduct operation) all the possible "correct parsings", that is, all available decompositions of an element into its possible building blocks.

We recall here some general facts about Hopf algebras that we will be using in the following.

A Hopf algebra $\mathcal{H}$ is a vector space over a field $\mathbb{K}$, endowed with

- a multiplication $m : \mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \to \mathcal{H}$;
- a unit $u : \mathbb{K} \to \mathcal{H}$;
- a comultiplication $\Delta : \mathcal{H} \to \mathcal{H} \otimes_{\mathbb{K}} \mathcal{H}$;
- a counit $\epsilon : \mathcal{H} \to \mathbb{K}$;
- an antipode $S : \mathcal{H} \to \mathcal{H}$

which satisfy the following properties. The multiplication operation is associative, namely the following diagram commutes

$$\begin{tikzcd}
\mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \arrow[r, "m\otimes id"] \arrow[d, "id\otimes m"] & \mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \arrow[d, "m"] \\
\mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \arrow[r, "m"] & \mathcal{H}
\end{tikzcd}$$

and the unit and multiplication are related by the commutative diagram

$$\begin{tikzcd}
& \mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \arrow[dd, "m"] & \\
\mathbb{K} \otimes_{\mathbb{K}} \mathcal{H} \arrow[ur, "u\otimes id"] \arrow[dr] & & \mathcal{H} \otimes_{\mathbb{K}} \mathbb{K} \arrow[ul, "id\otimes u"'] \arrow[dl] \\
& \mathcal{H} &
\end{tikzcd}$$

where the two unmarked downward arrows are the identifications induced by scalar multiplication. The coproduct is coassociative, namely the following diagram commutes

$$\begin{tikzcd}
\mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} & \mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \arrow[l, "\Delta\otimes id"'] \\
\mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \arrow[u, "id\otimes\Delta"] & \mathcal{H} \arrow[l, "\Delta"] \arrow[u, "\Delta"']
\end{tikzcd}$$

and the comultiplication and the counit are related by the commutative diagram

$$\begin{tikzcd}
 & \mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \arrow[dl, "\epsilon\otimes id"'] \arrow[dr, "id\otimes\epsilon"] & \\
\mathbb{K} \otimes_{\mathbb{K}} \mathcal{H} & \Delta & \mathcal{H} \otimes_{\mathbb{K}} \mathbb{K} \\
 & \mathcal{H} \arrow[ul] \arrow[uu] \arrow[ur] &
\end{tikzcd}$$

where the two unmarked upward arrows are the identifications given, respectively, by $x \mapsto 1_{\mathbb{K}} \otimes x$ and $x \mapsto x \otimes 1_{\mathbb{K}}$, with $1_{\mathbb{K}}$ the unit of the field $\mathbb{K}$. Moreover, Δ and ϵ are algebra homomorphisms, and m and u are coalgebra homomorphisms. The antipode $S : \mathcal{H} \to \mathcal{H}$ is a linear map such that the following diagram also commutes

$$\begin{tikzcd}
\mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \arrow[r, "m"] & \mathcal{H} & \mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \arrow[l, "m"'] \\
\mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \arrow[u, "id\otimes S"] & \mathcal{H} \arrow[l, "\Delta"] \arrow[r, "\Delta"'] \arrow[u, "u\circ\epsilon"] & \mathcal{H} \otimes_{\mathbb{K}} \mathcal{H} \arrow[u, "S\otimes id"]
\end{tikzcd}$$

In any graded bialgebra it is possible to construct an antipode map inductively by

$$S(X) = -X - \sum S(X')X'', \tag{2.1}$$

for any element X of the bialgebra, with coproduct $\Delta(X) = X \otimes 1 + 1 \otimes X + \sum X' \otimes X''$, where the X' and X'' are terms of lower degree. Thus, in the following we will focus only on the bialgebra structure and not discuss the antipode map explicitly.

Heuristically, what these properties of a Hopf algebra describe can be summarized as follows. As we will see in the specific example of trees below, as a vector space $\mathcal{H}$ consists of formal linear combinations of a specific class of objects (planar binary rooted trees, Feynman graphs, etc.). The operations will be defined on these generators and extended by linearity. The multiplication structure is a combination operation and the coproduct structure is a decomposition operation that lists all the possible different decompositions (parsings). The antipode is like a group inverse and it establishes the compatibility between multiplication and comultiplication, unit and counit.

2.2. **Binary rooted trees.** A rooted tree T is a finite graph whose geometric realization is simply connected (no loops), defined by a set of vertices V, with a distinguished element $v_r \in V$, the root vertex, and a set of edges E, which we can assume oriented with the uniquely defined orientation away from the root. The source and target maps $s, t : E \to V$ assign to each edge of the tree its source and target vertices. The leaves of the tree are the univalent vertices. They are also the sinks (no outgoing edges). A rooted tree is planar if it is endowed with an embedding of its geometric realization in the plane. Assigning planar embedding is equivalent to assigning a linear ordering of the leaves. The tree is vertex-decorated if there is a map $L_V : V \to D_V$ to a (finite) set. It is edge-decorated if similarly there is a map $L_E : E \to D_E$ to a finite set of possible edge decorations. We will assume the trees are vertex-decorated, and we will simply refer to them as decorated trees.

An *admissible cut* C on a rooted tree T is an operation that

(1) selects a number of edges of T with the property that every oriented path in T from the root to one of the leaves contains at most one of the selected edges
(2) removes the selected edges.

The result of an admissible cut is a disjoint union of a tree $\rho_C(T)$ that contains the root vertex and a forest $\pi_C(T)$, that is, a disjoint union $\pi_C(T) = \cup_i T_i$ of planar trees, where each T_i has a unique source vertex (no incoming edges), which we select as root vertex of T_i,

$$C(T) = \rho_C(T) \cup \pi_C(T). \tag{2.2}$$

An elementary admissible cut (or simply *elementary cut*) is a cut C consisting of a single edge.

2.3. **The Loday–Ronco Hopf algebra of binary rooted trees.** We note right away that this algebraic structure is different from that we considered in [37]. Since older forms of Minimalism, such as Stabler's version that we discuss here, incorporate linear ordering, it is necessary to work with binary rooted trees with an assigned choice of planar embedding (linear ordering of the leaves). This has immediate consequences on the type of algebraic structure that we need to work with, as the possible forms of product and coproduct operations are affected by the presence of this linear ordering. Another difference with respect to our setting in [37] is that in the new Minimalism, Merge acts on workspaces (a Hopf algebra of binary forests with no planar embeddings), while in the old Stabler Minimalism, Merge acts on (a Hopf algebra of) planar binary rooted trees.

Consider the vector space $\mathcal{V}_k$ spanned by the planar binary rooted trees T with k internal vertices (equivalently, with $k+1$ leaves). It has dimension

$$\dim \mathcal{V}_k = (\#D_V)^k \frac{(2k)!}{k!(k+1)!},$$

where $\#D_V$ is the cardinality of the set D_V of possible vertex labels. Let $\mathcal{V} = \oplus_{k\geq 0}\mathcal{V}_k$, with $\mathcal{V}_0 = \mathbb{Q}$. For a given label $d \in D_V$ the grafting operator $\wedge_d$ is defined as

$$\wedge_d : \mathcal{V} \otimes \mathcal{V} \to \mathcal{V}, \quad T_1 \otimes T_2 \mapsto T = T_1 \wedge_d T_2\,, \tag{2.3}$$

with $\wedge_d : \mathcal{V}_k \otimes \mathcal{V}_\ell \to \mathcal{V}_{k+\ell-1}$, that attaches the two roots v_{r_1} of T_1 and v_{r_2} of T_2 to a single root vertex v labelled by $d \in D_V$.

One also introduces the following associative concatenation operations on planar binary rooted trees: given S and T, the tree $S\backslash T$ (S under T) is obtained by grafting the root of T to the rightmost leaf of S, while T/S (S over T) is the tree obtained by grafting the root of T to the leftmost leaf of S. The grafting operation (2.3) is related to these concatenations by

$$T_1 \wedge_d T_2 = T_1/S\backslash T_2, \tag{2.4}$$

where S is the planar binary tree with a single vertex decorated by $d \in D_V$. Each planar binary rooted tree is described as a grafting $T = T_\ell \wedge_d T_r$, of the trees stemming to the left and right of the root vertex. We remove the explicit mention of the vertex decorations when not needed.

The Loday–Ronco Hopf algebra $\mathcal{H}_{LR}$ of planar binary rooted trees is obtained from the vector space $\mathcal{V}$ by defining a multiplication and a comultiplication inductively by degrees. For trees $T = T_\ell \wedge T_r$ and $T' = T'_\ell \wedge T'_r$ the product defined in [35] can be built inductively using the property

$$T \star T' = T_\ell \wedge (T_r \star T') + (T \star T'_\ell) \wedge T'_r,$$

with the tree consisting of a single root vertex $\bullet$ as the unit. The coproduct similarly can be built from lower degree terms by the property

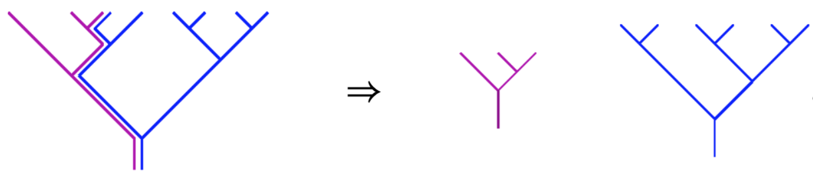
$$\Delta(T) = \sum_{j,k} (T_{\ell,j} \star T_{r,k}) \otimes (T'_{\ell,n-j} \wedge T'_{r,m-k}) + T \otimes \bullet$$

where $T = T_\ell \wedge T_r$ and $\Delta(T_\ell) = \sum_j T_{\ell,j} \otimes T'_{\ell,n-j}$ and $\Delta(T_r) = \sum_k T_{r,k} \otimes T'_{r,m-k}$, for $T_\ell \in \mathcal{V}_n$ and $T_r \in \mathcal{V}_m$.

In [35] this Hopf algebra $\mathcal{H}_{LR}$ of planar binary rooted trees is described in terms of the Hopf algebra structure on the group algebra $\mathbb{Q}[S_\infty] = \oplus_n \mathbb{Q}[S_n]$ of the symmetric group. Namely, the inclusion of the Hopf algebra of non-commutative symmetric functions in the Malvenuto-Reutenauer Hopf algebra of permutations factors through the Loday-Ronco Hopf algebra of planar binary rooted trees, see also [1]. In [7] a version of the Loday-Ronco Hopf algebra of planar binary rooted trees was used for the renormalization of massless quantum electrodynamics and and explicit isomorphisms between the Loday-Ronco Hopf algebra of planar binary rooted trees and the (noncommutative) Connes–Kreimer Hopf algebra of renormalization were constructed in [1], [25], [20].

2.4. **Graphical form of coproduct and product.** It is shown in [1] that the coproduct and product of the Loday-Ronco Hopf algebra of planar binary rooted trees can be conveniently visualized in the following way.

Given a tree T, one subdivides it into two parts by cutting along the path from one of the leaves to the root (illustration from [1])

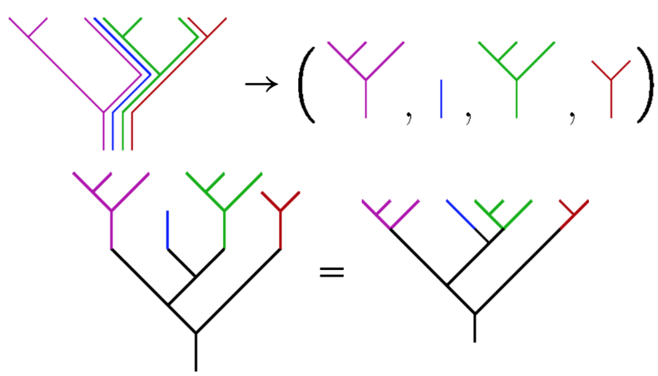


and the coproduct then takes the form of a sum over all such decompositions

$$\Delta(T) = \sum T' \otimes T'' \tag{2.5}$$

where T', T'' are, respectively the parts of T to the left and right of the path from a leaf to the root and all choices of leaves are summed over.

In order to form the product $T_1 \star T_2$, suppose that T_1 has $n_1 + 1$ leaves and T_2 has $n_2 + 1$ leaves. Consider again subdivisions of the tree T_1 obtained by cutting along paths from the leaves to the root, including trees consisting of just the path itself, with the cuts chosen so as to obtain $n_2 + 1$ subtrees of T_1 (illustration from [1] in a case with $n_1 = 5$, $n_2 = 3$)



We write $\mathcal{V}^{(k)}$ for the $\mathbb{Q}$-vector space spanned by the planar binary rooted trees with k leaves, and we write the usual operadic composition operation of grafting roots to leaves as above in the form

$$\gamma : \mathcal{V}^{(k_0)} \times \cdots \times \mathcal{V}^{(k_n)} \times \mathcal{V}^{(n+1)} \to \mathcal{V}^{(k_0+\cdots+k_n)} \tag{2.6}$$

where $\gamma(T_0, \ldots, T_n; T)$ is the tree obtained by grafting the trees $T_i \in \mathcal{V}^{(k_i)}$ in collection $(T_0, \ldots, T_n)$ to the tree $T \in \mathcal{V}^{(n+1)}$, by attaching the root of T_i to the i-th leaf of T, as illustrated above.

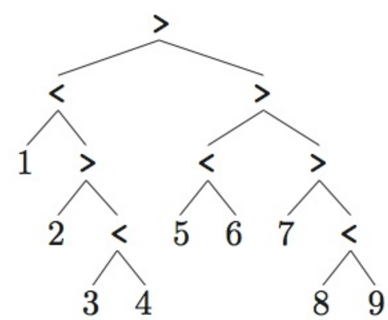
The product in $\mathcal{H}_{LR}$ is then written in the form

$$T \star T' = \sum_{(T_0,\ldots,T_n)} \gamma(T_0,\ldots,T_n;T') \tag{2.7}$$

with $n+1$ the number of leaves of T', and the sum over subdivisions into subtrees extracted according to the rule described above.

2.5. **Labelled trees.** The previous subsections only dealt with general mathematical formalism about planar binary rooted trees. We now consider more specifically the case of Stabler's Computational Minimalism.

In Stabler's formulation, one considers planar binary rooted trees such as the following example (from [42]).



where the ordering of the leaves determines and is determined by the planar embedding of the tree, and the labels at the inner vertices serve the purpose of pointing toward the head.

More precisely, all internal vertices and the root vertex are labelled by symbols in the set $\{>,<\}$. The purpose of the labels $>$ and $<$ is to identify where the *head* of the tree is: in the example above the head is the leaf vertex number 8. In addition to these labels, one also considers a finite set of syntactic features $X \in \{N,V,A,P,C,T,D,\ldots\}$ and "selector" features denoted by the symbol σX for a head selecting a phrase XP. More generally we can have labels that are strings (ordered finite sets) $\alpha = X_0X_1\cdots X_r$ of syntactic features as above. We also consider labels given by letters ω and $\bar{\omega}$ that stand for "licensor" and "licensee". (Note: in [42] the notation $=X$ is used for feature selector instead of our σX, and the notation $\pm X$ is used for licensor/licensee pairs, but we prefer to avoid using mathematical notation with a meaning different from its generally accepted one, to avoid confusion, hence we will use the letters σ and $\omega, \bar{\omega}$ instead.)

We write the result of external merge applied to constituents α and β as in [42], as the labelled planar binary rooted tree

$$\overset{<}{\overbrace{\alpha \quad \beta}} \qquad \text{rather than} \qquad \overset{\alpha}{\overbrace{\alpha \quad \beta}}$$

This means that we consider as generators of the Loday–Ronco Hopf algebras the binary trees with labels that include the syntactic features, as well as symbols $<$ and $>$, as in [42], used to point towards the head when merge is applied.

By (2.4) it is clear that the head of the tree $T_1 \wedge_> T_2$ is the head of T_2 and the head of the tree $T_1 \wedge_< T_2$ is the head of T_1.

Recall also that a *maximal projection* in T is a subtree of T that is not a proper subtree of any larger subtree with the same head. As pointed out in [42], in the example illustrated above, the leaves $\{2,3,4\}$ determine a subtree with head vertex the leaf numbered 3, but any larger subtree in T would have a different head, hence this subtree is a maximal projection. So is the subtree determined by the leaves $\{5, 6\}$, for instance.

We will write $\mathcal{H}_{ling}$ for the Loday-Ronco Hopf algebra $\mathcal{H}_{LR}$ with the choice of labels described above. We will write $\mathcal{V}_{ling}$ for the underlying $\mathbb{Q}$-vector space spanned by the binary trees with the labeling described above.

As we pointed out already, the choice of working with *planar* binary rooted trees corresponds to an assigned linear ordering of its leaves, which in turn is the linear ordering of the resulting sentence when spoken or read. Thus, by working with planar trees we are assuming that linear ordering is determined for the result of Merge. This is the same assumption made in [42]. This is the crucial point in the comparison with the new Minimalism, and we will discuss it more explicitly in Section 3 below.

2.6. **Old formulation of External and Internal Merge.** We now discuss the external and internal merge operations in the old formulations of minimalism. We first describe the operations at the level of the underlying combinatorial tree, and then we give the more precise constraints on when the operations can be applied, based on the labels of the trees involved.

Thus, the way we should view external and internal merge is as *partially defined operations*

$$\mathcal{E} : \mathcal{V}_{ling} \otimes \mathcal{V}_{ling} \to \mathcal{V}_{ling}$$
$$\mathcal{I} : \mathcal{V}_{ling} \to \mathcal{V}_{ling}$$

where we assume that the operations defined on generators are extended by (bi)linearity, so that they are defined on linear subspaces

$$\mathrm{Dom}(\mathcal{E}) \subset \mathcal{V}_{ling} \otimes \mathcal{V}_{ling}$$
$$\mathrm{Dom}(\mathcal{I}) \subset \mathcal{V}_{ling}$$

that we will describe more precisely below.

At the level of the underlying combinatorial trees external and internal merge are simply defined as the following operations.

The combinatorial structure of the external merge is given by

$$\mathcal{E}(T_1 \otimes T_2) = \begin{cases} \bullet \wedge T_2 & T_1 = \bullet \\ T_2 \wedge T_1 & \text{otherwise,} \end{cases} \tag{2.8}$$

where $\bullet$ denotes the tree consisting of a single vertex, and the $\wedge$ operation is the grafting operation $\wedge_d$ defined as in (2.3), where the specific label $d \in \{>, <\}$ according to a rule that will be discussed more in detail in §2.7 below.

The combinatorial structure of the internal merge is given by

$$\mathcal{I}(T) = \pi_C(T) \wedge \rho_C(T), \tag{2.9}$$

where C is an elementary admissible cut of T with $\rho_C(T)$ the remaining pruned tree that contains the root of T and $\pi_C(T)$ the part that is severed by the cut (which in the case of an elementary cut is itself a tree rather than a forest). Again $\wedge$ is as in (2.3), with a label that will be made more precise below, and the choice of the elementary cut will also depend on conditions on tree labels, see §2.8.

We now look more precisely at the structure of external and internal merge and at their domains of definition. It should be noted how working with planar structures and with conditions of matching/related labels makes the underlying mathematical operations more difficult to describe, as they are only defined on specific domains and with additional conditions.

2.7. **Old form of External Merge.** The more precise definition of the External Merge in the older forms of Miimalism is given as follows (where we are adapting the description of [42] to our terminology and notation).

As in [42], we use the notation $T[\alpha]$ for a tree where the head is labelled by an ordered set of syntactic features starting with α. The label α consists of a string of syntactic features of the form

$$\alpha = X_0 X_1 \cdots X_r \quad \text{or} \quad \alpha = \sigma X_0 X_1 \cdots X_r,$$

with σ the selector symbol.

We introduce the notation $\hat\alpha$ for the string obtained from α after removing the first feature (other than the selection symbol). Namely, for $\alpha = X_0 X_1 \cdots X_r$ or $\alpha = \sigma X_0 X_1 \cdots X_r$, we have $\hat\alpha = X_1 \cdots X_r$.

The external merge $T = \mathcal{E}(T_1[\sigma\alpha], T_2[\alpha])$ of two trees $T_1[\sigma\alpha]$ and $T_2[\alpha]$ is given by

$$\mathcal{E}(T_1[\sigma\alpha], T_2[\alpha]) = \begin{cases} T_1[\widehat{\sigma\alpha}] \wedge_< T_2[\hat\alpha] & |T_1| = 1 \\ T_2[\hat\alpha] \wedge_> T_1[\widehat{\sigma\alpha}] & |T_1| > 1. \end{cases} \tag{2.10}$$

This clearly agrees with (2.8) at the level of the underlying combinatorial trees. The main difference with (2.8) is that here the operation can be performed only if the heads are decorated with syntactic features $\sigma\alpha$ and α, respectively. Thus the domain of definition $\mathrm{Dom}(\mathcal{E}) \subset \mathcal{V}_{ling} \otimes \mathcal{V}_{ling}$ of the external merge operation is the vector space on the set of generators

$$\mathrm{Dom}(\mathcal{E}) = \mathrm{span}_{\mathbb{Q}}\{(T_1[\beta], T_2[\alpha]) \,|\, \beta = \sigma\alpha\}. \tag{2.11}$$

2.8. **Old version of Internal Merge.** In the older formulations of Minimalism, one considers a second Merge operation, considered as distinct from External Merge, namely the *Internal Merge*. We again adapt to our notation and terminology the formulation given in [42]. We use the same notation as above for $T[\alpha]$ with $\alpha = X_0 X_1 \ldots X_r$ or $\alpha = \sigma X_0 X_1 \ldots X_r$ a string of syntactic features possibly starting with a selector, and we write $\hat\alpha = X_1 \ldots X_r$. The internal merge is again modeled on the basic grafting operation of planar rooted binary trees $\wedge_d$, but this time its input is a single tree $T[\alpha]$.

Given a planar binary rooted tree T containing a subtree T_1, and given another binary rooted tree T_2, we denote as in [42] by $T\{T_1 \to T_2\}$ the planar binary rooted tree obtained by removing the subtree T_1 from T and replacing it with T_2. In particular, we write $T\{T_1 \to \emptyset\}$ for the tree obtained by removing the subtree T_1 from T. Note that, in order to ensure that the result of this operation is still a tree, we need to assume that if an internal vertex of T belongs to the subtree T_1, then all oriented paths in T from this vertex to a leaf also belong to T_1, where paths in the tree are oriented from root to leaves. We only consider subtrees that have this property. When it is necessary to stress this fact, we will refer to such subtrees as *complete subtrees*. Note that these are exactly the subtrees that can be obtained by applying a single elementary admissible cut C to the tree T. The tree $T\{T_1 \to \emptyset\}$ is then the same as the tree $\rho_C(T)$.

Consider a tree $T[\alpha]$ where $\alpha = X_0 \cdots X_r$ or $\alpha = \sigma X_0 \cdots X_r$ or $\alpha = \omega X_0 \cdots X_r$ or $\alpha = \bar\omega X_0 \cdots X_r$.

The domain $\mathrm{Dom}(\mathcal{I}) \subset \mathcal{V}_{ling}$ of the internal merge is given by the subspace

$$\mathrm{Dom}(\mathcal{I}) = \mathrm{span}_{\mathbb{Q}}\{T[\alpha] \,|\, \exists T_1[\beta] \subset T[\alpha], \text{with } \beta = \bar\omega X_0 \hat\beta,\ \alpha = \omega X_0 \hat\alpha\}\,, \tag{2.12}$$

where $T_1 \subset T$ is a subtree (in the sense specified above), and

$$\mathcal{I}(T[\alpha]) = T_1^M[\hat\beta] \wedge_> T\{T_1[\beta]^M \to \emptyset\} = \pi_C(T) \wedge_> \rho_C(T), \tag{2.13}$$

where C is the elementary admissible cut specified by the subtree T_1^M (the maximal projection of the head of T_1) and the condition (given by the matching labels ωX_0 and $\bar{\omega} X_0$) that $T[\alpha] \in \mathrm{Dom}(\mathcal{I})$. The head of $\pi_C(T) \wedge_> \rho_C(T)$ gets the label $\hat{\alpha}$.

Stabler's External and Internal Merge, that we recalled here in (2.10) and (2.13), have issues at the linguistics level, such as problems with potentially unlabelable exocentric constructions. For example (as observed by Riny Huijbregts) Internal Merge as in (2.13) gives $\{XP, YP\}$ results, and labeling cannot be assigned, unless a further application of Internal Merge takes XP or YP in a criterial position, where structural agreement, SPEC-Head agreement, can take place between specifier and head. In [42], two further variants of External and Internal Merge are introduced in order to deal with "persistent features". This results in what is referred to in [42] as "conflated minimalist grammars" (CMGs). These additional forms of External and Internal Merge, however, do not solve the problem mentioned above and further complicate the structure, so we will focus here on analyzing the algebraic structure determined by just the forms (2.10) and (2.13) of Stabler's External and Internal Merge.

2.9. **Old version of Internal Merge and the coalgebra structure.** As we have discussed above, the internal merge is only partially defined on $\mathcal{V}_{ling}$ because it requires the existence of matching conditions on the label, which determine the domain $\mathrm{Dom}(\mathcal{I}) \subset \mathcal{V}_{ling}$. We now discuss now how the domain $\mathrm{Dom}(\mathcal{I})$ and the internal merge operation $\mathcal{I}$ behave with respect to the coproduct of the Hopf algebra $\mathcal{H}_{ling}$.

Consider the coproduct $\Delta(T)$ of a tree $T \in \mathrm{Dom}(\mathcal{I})$. This is given by the sum of decompositions $\Delta(T) = \sum T' \otimes T''$ as illustrated in (2.5). In each of these decompositions one side will contain the head of the tree T (or possibly both sides if the leaf used to cut the tree happens to be also the head).

If both the head of T and the head of $\pi_C(T)$ are contained in the same side of the partition, then that side still belongs to $\mathrm{Dom}(\mathcal{I})$. Thus, these pieces of the coproduct are in $\mathrm{Dom}(\mathcal{I}) \otimes \mathcal{H}_{ling} + \mathcal{H}_{ling} \otimes \mathrm{Dom}(\mathcal{I})$. The remaining terms, with the heads of T of $\pi_C(T)$ on different sides of the decomposition only belong in general to $\mathcal{H}_{ling} \otimes \mathcal{H}_{ling}$ and fall outside of the domain of the internal merge.

This suggests a modification of the coproduct on $\mathcal{H}_{ling}$ with respect to the usual Loday-Ronco coproduct (2.5). For a generator T that is not in $\mathrm{Dom}(\mathcal{I})$ we just set $\pi_C(T) = T$, that is, no cut is performed.

For $T \in \mathrm{Dom}(\mathcal{I})$, let $h(T)$ and $h(\pi_C(T))$ be the leaves of T that are the head of T and the head of $\pi_C(T)$, respectively. If T is in $\mathrm{Dom}(\mathcal{I})$ then C is, as above, the elementary admissible cut determined by the label conditions. Let $\mathcal{P}_{\mathcal{I}}(T)$ denote the set of bipartitions

(2.14)
$$\mathcal{P}_{\mathcal{I}}(T) = \{T = (T', T'') \,|\, (h(T) \in T' \text{ and } h(\pi_C(T)) \in T') \text{ or } (h(T) \in T'' \text{ and } h(\pi_C(T)) \in T'')\}\,.$$

For trees outside the domain $\mathrm{Dom}(\mathcal{I})$ the set $\mathcal{P}_{\mathcal{I}}(T)$ just counts all bipartitions as we have set $\pi_C(T) = T$.

We then define the modified coproduct

$$\Delta_{\mathcal{I}}(T) := \sum_{(T',T'') \in \mathcal{P}_{\mathcal{I}}(T)} T' \otimes T'', \tag{2.15}$$

which is the same as (2.5) outside of $\mathrm{Dom}(\mathcal{I})$.

With this modified coproduct $\mathrm{Dom}(\mathcal{I})$ is a *coideal* of the coalgebra $\mathcal{H}_{ling}$, namely

$$\Delta_{\mathcal{I}}(\mathrm{Dom}(\mathcal{I})) \subset \mathrm{Dom}(\mathcal{I}) \otimes \mathcal{H}_{ling} + \mathcal{H}_{ling} \otimes \mathrm{Dom}(\mathcal{I})\,. \tag{2.16}$$

2.10. **Old version of Internal Merge and the algebra structure.** We now discuss the behavior of internal merge with respect to the product structure of $\mathcal{H}_{ling}$. In this case, arguing in a similar way as for the coproduct above, there is a modification $\star_{\mathcal{I}}$ of the original product of $(\mathcal{H}_{LR}, \star)$ that remains the same outside of $\mathrm{Dom}(\mathcal{I})$ and takes into account the additional data in $\mathrm{Dom}(\mathcal{I})$ and that has the effect of making $\mathrm{Dom}(\mathcal{I})$ into a *right ideal* with respect to the algebra $(\mathcal{H}_{ling}, \star_{\mathcal{I}})$. However, $\mathrm{Dom}(\mathcal{I})$ is not a left ideal.

We define the modified product $\star_{\mathcal{I}}$ on $\mathcal{H}_{ling}$ in the following way. Let $h(T)$ denote the head of T. Given trees T, T' where T' has $n+1$ leaves, consider the set $\mathcal{P}_{\mathcal{I}}(T, T')$ given by decompositions $(T_0, \dots, T_n)$ of T as above with

$$\mathcal{P}_{\mathcal{I}}(T, T') := \{(T_0, \dots, T_n) \,|\, h(T) \text{ and } h(\pi_C(T)) \in T_{h(T')}\} \tag{2.17}$$

Namely those decompositions $(T_0, \dots, T_n)$ with the property that the head of T (and the head of $\pi_C(T)$ in the case where $T \in \mathrm{Dom}(\mathcal{I})$) lies in the component $T_{h(T')}$ that is grafted to the head of T'. In the case of $T \notin \mathrm{Dom}(\mathcal{I})$ the condition reduces to just $h(T) \in T_{h(T')}$. Note that it is always possible to have such decompositions, as some of the components T_i can always be taken to be copies of just the path from a leaf to the root, so that the relevant piece of the decomposition can be placed at the $h(T')$-position.

We then take the product $\star_{\mathcal{I}}$ on $\mathcal{H}_{ling}$ of the form

$$T \star_{\mathcal{I}} T' = \sum_{(T_0, \dots, T_n) \in \mathcal{P}_{\mathcal{I}}(T, T')} \gamma(T_0, \dots, T_n; T') \,. \tag{2.18}$$

The head of each $\gamma(T_0, \dots, T_n; T')$ is then the same as the head of T, which we write in shorthand as $h(T \star_{\mathcal{I}} T') = h(T)$. Moreover, by construction the component $T_{h(T')}$ is in $\mathrm{Dom}(\mathcal{I})$ when $T \in \mathrm{Dom}(\mathcal{I})$, so that $T \star_{\mathcal{I}} T'$ is itself in $\mathrm{Dom}(\mathcal{I})$. This shows that $\mathrm{Dom}(\mathcal{I})$ is a right ideal with respect to the algebra $(\mathcal{H}_{ling}, \star_{\mathcal{I}})$,

$$\mathrm{Dom}(\mathcal{I}) \star_{\mathcal{I}} \mathcal{H}_{ling} \subset \mathrm{Dom}(\mathcal{I}) \,.$$

It is, however, not a left ideal.

For $T \in \mathrm{Dom}(\mathcal{I})$ we then have

$$\mathcal{I}(T \star_{\mathcal{I}} T') = \sum_{(T_0, \dots, T_n) \in \mathcal{P}_{\mathcal{I}}(T, T')} \pi_C(T_{h(T')}) \wedge_{>} \gamma(T_0, \dots, \rho_C(T_{h(T')}), \dots, T_n; T') \,. \tag{2.19}$$

Combining the behavior with respect to the product with the previous observation about the coproduct we conclude that the internal merge $\mathcal{I}$ defines a right $(\mathcal{H}_{ling}, \star_{\mathcal{I}})$-module given by the cosets

$$\mathcal{M}_{\mathcal{I}} := \mathrm{Dom}(\mathcal{I}) \backslash \mathcal{H}_{ling}$$

where $\mathcal{M}_{\mathcal{I}}$ is also a coalgebra with the coproduct induced by $(\mathcal{H}_{ling}, \Delta_{\mathcal{I}})$.

2.11. **Iterated Internal Merge.** In fact, the construction above can be extended to a nested family of right-ideal coideals, given by the domains of the iterations of the internal merge, $\mathcal{D}_{N+1} \subset \mathcal{D}_N$ with

$$\mathcal{D}_N := \mathrm{Dom}(\mathcal{I}^N) \,.$$

When we consider repeated application of N internal merge operations, starting from a given tree $T[\alpha]$, where α is a finite list of features each of which can be either X_i or ωX_i of $\bar{\omega} X_i$ as above, we need to assume the following conditions

(1) There are N subtrees $T_1, \dots, T_N$ in T, each of which is a complete subtree, in the sense specified above.

(2) Let $T_1^M, \ldots, T_N^M$ be the maximal projections of the subtrees, in the sense recalled above. These are also complete subtrees of T.
(3) The subtrees T_i^M are disjoint.

In addition to these combinatorial conditions, we have a matching labels condition that specifies the domain of the composite operation. Let $T[\alpha]$ be the given tree as a labelled tree, and let $T_1[\beta^{(1)}], \ldots, T_N[\beta^{(N)}]$ be the labelled subtrees, with the conditions listed above. The domain of the composite internal merge is then given by
(2.20)

$$\mathrm{Dom}(\mathcal{I}^N) = \left\{ T[\alpha] \,\Big|\, \exists T_1[\beta^{(1)}], \ldots, T_N[\beta^{(N)}] \text{ with } \left\{ \begin{array}{l} (1), (2), (3) \text{ are satisfied} \\ \beta_0^{(1)} = \bar{\omega} X_0, \ldots, \beta_0^{(N)} = \bar{\omega} X_{N-1} \\ \alpha = \omega X_0 \omega X_1 \cdots \omega X_{N-1} \cdots \end{array} \right\} \right\}.$$

For a forest $F = T_1 \cdots T_\ell$, we define a grafting operation $\wedge^\ell$ that consists of the repeated application of the grafting $\wedge$ to the trees in F,

$$\bigwedge^{\ell} F = T_1 \wedge T_2 \wedge \cdots \wedge T_\ell. \tag{2.21}$$

The conditions (1), (2), and (3) above ensure that the choice of the subtrees $T_1^M, \ldots, T_N^M$ corresponds to an admissible cut C of the tree T, with the number of cut branches $\#C = N$ The planar binary rooted tree obtained by this operation is then

$$\mathcal{I}^{\#C}(T[X]) = \bigwedge^{1+\#C} \left(\pi_C(T)[\hat{\mathbb{Y}}] \ \rho_C(T)[\hat{X}^N] \right), \tag{2.22}$$

where we use the notation $\pi_C(T)[\hat{\mathbb{Y}}]$ for the forest $\pi_C(T) = T_N^M \cdots T_1^M$, where the label $[\hat{\mathbb{Y}}]$ means

$$\pi_C(T)[\hat{\mathbb{Y}}] = T_N^M[\hat{\beta}^{(N)}] \cdots T_1^M[\hat{\beta}^{(1)}]$$

and the label $[\hat{\alpha}^N]$ of the tree $\rho_C(T)$ stands for what remains of the original label X after the initial terms $\omega X_0 \omega X_1 \cdots \omega X_{N-1}$ are removed.

Each $\mathcal{D}_{N+1} \backslash \mathcal{D}_N$ determines a coideal in the coalgebra $\mathcal{D}_{N+1} \backslash \mathcal{H}_{ling}$ and this gives a projective system of right-module coalgebras

$$\mathcal{M}_{\mathcal{I}^N} := \mathrm{Dom}(\mathcal{I}^N) \backslash \mathcal{H}_{ling}\,.$$

2.12. **Coideals, recursive structures, and symmetries.** In the Minimalist Model of syntax, the Merge operator is the main mechanism for the construction of recursive and hierarchical structures. So indeed it is not surprising that the Hopf-algebraic formalism occurs naturally in this context, as it is known to describe similar hierarchical structures (such as Feynman graphs) in fundamental physics. In the applications of Hopf algebras to physics, especially to renormalization in perturbative quantum field theories, Hopf ideals are closely related to the recursive implementation of symmetries (Ward identities) and the recursive construction of solutions to equations of motion (Dyson-Schwinger equations), [19], [45].

We have discussed in our previous paper [37] how the algebraic formulation of the new Minimalism closely resembles the mathematical structure of Dyson-Schwinger equations in physics. However, the analogous mathematical structure describing Merge in the old forms of Minimalism differs significantly from that basic fundamental formulation we have in the case of the new Minimalism.

The main difference, as shown above, lies in the intrinsic asymmetry of the old form of the Merge operation, which only gives rise to a weaker structure than Hopf ideals, namely the right-ideal coideals discussed in §2.10. Thus, instead of a quotient Hopf algebra as in the case of implementation of symmetries in quantum field theory, one only obtains a quotient right-module coalgebra.

Such objects, quotient right-module coalgebras, sometimes referred to as "generalized quotients" of Hopf algebras, do provide a suitable notion of quotients in the case of noncommutative Hopf algebras, [38], [43]. They are also studied in the context of the theory of Hopf–Galois extensions, designed to provide good geometric analogs in noncommutative geometry of principal bundles and torsors in ordinary geometry, see for instance [41]. However, the type of structure involved, in order to accommodate the requirement of working with planar trees, becomes significantly more complicated than in the case of free symmetric Merge of the new Minimalism.

We can think of the right-module coalgebras $\mathcal{M}_{\mathcal{I}^N}$ as a family of geometric spaces (in a noncommutative sense) that implement the recursive structures of syntax generated by the Merge operation, in a sense similar to how quotients by Hopf ideals in physics recursively implement the gauge symmetries or the recursively constructed solutions to the equations of motion. This, however, is a significantly weaker (and at the same time significantly more complicated) algebraic structure than the one we see occurring in the newer version of Minimalism, as shown in [37] and discussed in §3 below.

2.13. **Partial operated algebra.** We now discuss more in detail the mathematical structure of the old formulation of External Merge, and we show how it creates an independent structure of a very different nature from Internal Merge discussed in §2.10. These very different mathematical formulations of Internal and External Merge are another significant drawback of the older Minimalism in comparison with the newer.

We can consider the data $(\mathcal{H}_{ling}, \star_{\mathcal{I}}, \mathrm{Dom}(\mathcal{I}), \mathcal{I})$ discussed above as a generalization of the notion of *operated algebra* (see [24], [46]) where one considers data $(\mathcal{A}, \star, F)$ of an algebra together with a linear operator $F : \mathcal{A} \to \mathcal{A}$. Here we allow for the case where the linear operator $F : \mathrm{Dom}(F) \to \mathcal{A}$ is only defined on a smaller domain $\mathrm{Dom}(F) \subset \mathcal{A}$ which is a right ideal of $\mathcal{A}$. We can refer to this structure $(\mathcal{H}_{ling}, \star_{\mathcal{I}}, \mathrm{Dom}(\mathcal{I}), \mathcal{I})$ as a *partial operated algebra*.

The setting of operated algebras was introduced by Rota [40] as a way of formalizing various instances of linear operators $F : \mathcal{A} \to \mathcal{A}$ on an algebra that satisfy polynomial constraints. The simplest example of such polynomial constraints is the identity $F(a \star b) = F(a) \star F(b)$ that makes F an actual algebra homomorphism. Another example is the Leibniz rule constraint $F(a \star b) = a \star F(b) + F(a) \star b$ that makes F a derivation. Other interesting polynomial constraints are Rota–Baxter relations, $F(a) \star F(b) = F(a \star F(b)) + F(F(a) \star b) + \lambda F(a \star b)$, which depending on the value of the parameter λ can represent various types of operations such as integration by parts, or extraction of the polar part of a Laurent series, and play an important role in the Hopf algebra formulation of renormalization in physics.

In the case of the internal merge we considered above, not only the linear operator is only partially defined as $\mathcal{I} : \mathrm{Dom}(\mathcal{I}) \to \mathcal{H}_{ling}$, but the identity (2.19) that expresses the internal merge of a product is not directly of a simple poynomial form as in Rota's operated algebra program. However, it appears to be an interesting question whether the type of relation (2.19) can be accommodated in terms of the approach to Rota's program via *term rewriting systems* (in the sense of [2]) developed in [22], [24].

2.14. **Old version of External merge and the Hopf algebra structure.** We now focus on the binary operation $\mathcal{E}$ of External Merge, defined on the subdomain $\mathrm{Dom}(\mathcal{E}) \subset \mathcal{H}_{ling} \otimes \mathcal{H}_{ling}$ described in (2.11). The structure of external merge is simpler than the internal merge discussed above, and closely related to the usual Hochschild cocycle that defines the grafting operations in the Hopf-algebraic construction of Dyson-Schwinger equations in physics.

The "operated algebra" viewpoint we discussed briefly above is especially useful in analyzing the external merge, along the lines of extending the notion of operated algebra from unitary to binary operations discussed in [46].

We recall from [46] the notion of operated algebra based on binary operations. These structure is called a $\vee_\Omega$-algebra in [46]. In our notation we use $\wedge$ instead of $\vee$ for the grafting of binary trees used in the merge operation, following the convention of writing syntactic parsing trees with the root at the top and the leaves at the bottom. So we are going to refer to this structure as $\wedge_\Omega$-algebra.

A $\wedge_\Omega$-algebra is an algebra $(\mathcal{A}, \star)$ together with a family of binary operations $\{\wedge_\alpha\}_{\alpha\in\Omega}$ satisfying the identity

$$a \star b = a_1 \wedge_\alpha (a_2 \star b) + (a \star b_1) \wedge_\alpha b_2 \,, \tag{2.23}$$

for $a = a_1 \wedge_\alpha a_2$ and $b = b_1 \wedge_\alpha b_2$. A $\wedge_\Omega$-bialgebra (or Hopf algebra) $(\mathcal{H}, \Delta, \star, \wedge_\Omega)$ is a bialgebra (or Hopf algebra) that is also a $\wedge_\Omega$-algebra. It is a *cocycle* $\wedge_\Omega$-Hopf algebra if the binary operations $\wedge_\Omega$ also satisfy the cocycle identity

$$\Delta(a \wedge_\alpha b) = (a \wedge_\alpha b) \otimes 1 + (\star \otimes \wedge_\alpha) \circ \tau(\Delta(a) \otimes \Delta(b)) \,, \tag{2.24}$$

where $\tau : \mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H} \to \mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H} \otimes \mathcal{H}$ is the permutation that exchanges the two middle factors.

It is shown in [46] that the Loday-Ronco Hopf algebra $\mathcal{H}_{LR}$ of planar binary rooted trees with the binary operations $\wedge_\alpha$ that graft two trees T_1, T_2 as the left and right subtrees at a new binary root with label α is a cocyle $\wedge_\Omega$-Hopf algebra. The range of $\wedge_\alpha$ is a coideal in $\mathcal{H}_{LR}$. It is also shown in [46] that the $\wedge_\Omega$-Hopf algebra $\mathcal{H}_{LR}$ is the free cocycle $\wedge_\Omega$-Hopf algebra, that is, the initial object in the category of cocycle $\wedge_\Omega$-Hopf algebras.

The external merge operation is modelled on the grafting operations $\wedge_\alpha$ of the Loday-Ronco Hopf algebra, with labels $\alpha \in \{<, >\}$. The main differences are that the external merge inverts the order when the first tree is nontrivial,

$$\mathcal{E}(T_1, T_2) = T_2 \wedge_> T_1 \quad \text{for } T_1 \neq \bullet$$

$$\mathcal{E}(\bullet, T_2) = \bullet \wedge_< T_2$$

and the fact that the operation is only defined on a domain $\mathrm{Dom}(\mathcal{E}) \subset \mathcal{H}_{ling} \otimes \mathcal{H}_{ling}$ of pairs of trees $(T_1[\beta], T_2[\alpha])$ with matching labels $\beta = \sigma\alpha$.

Both the $\wedge_\alpha$-algebra identity (2.23) and the cocycle identity (2.24) are still satisfied, whenever all the terms involved are in the domain of the merge operator. This leads naturally to consider external merge as a structure of *partially defined cocycle $\wedge_\Omega$-Hopf algebra*.

Namely, a partially defined cocycle $\wedge_\Omega$-bialgebra (or Hopf algebra) is a bialgebra (or Hopf algebra) $(\mathcal{H}, \Delta, \star)$ together with a family $\{\wedge_\alpha\}_{\alpha\in\Omega}$ of binary operations defined on linear subspaces $\mathrm{Dom}(\wedge_\alpha) \subset \mathcal{H} \otimes \mathcal{H}$,

$$\wedge_\alpha : \mathrm{Dom}(\wedge_\alpha) \hookrightarrow \mathcal{H} \otimes \mathcal{H} \to \mathcal{H}$$

that satisfy (2.23) and (2.24), whenever all the terms are in $\mathrm{Dom}(\wedge_\alpha)$.

The Hopf algebra $\mathcal{H}_{ling}$ with the external merge operator $\mathcal{E}$ is then a partially defined cocycle $\wedge_\Omega$-Hopf algebra.

3. New Minimalism, Merge, magmas and Hopf algebras

The new version of Minimalism, as presented in [10], [11], [13], and in our mathematical formulation in [37], avoids all the complications arising in the previous formulation, both coming from the presence of planar structures and from the need for labeling and projection. By delegating these aspects to a later externalization procedure, the core computational mechanism of Merge becomes very transparent and simple from the mathematical perspective and no longer leads to very different structures underlying Internal and External Merge. The formulation is also no longer plagued by the problem of partially defined operations and corresponding domains.

The whole mathematical structure of Merge in the new version of Minimalism is discussed in detail in our paper [37], including a proposed model for externalization. We summarize it here for direct comparison with the old Minimalism discussed in the previous section.

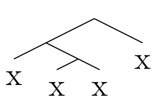
3.1. The core computational structure. In the new version of Minimalism, one starts with a core computational structure, which is the recursive construction of binary rooted trees, where trees are now just abstract trees, not endowed with planar structure. This structure is described mathematically in the following way.

The set $\mathfrak{T}$ of finite binary rooted trees without planar structure is obtained as the *free non-associative commutative magma* whose elements are the balanced bracketed expressions in a single variable x, with the binary operation (binary set formation)

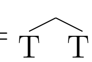
$$(\alpha,\beta)\mapsto \mathfrak{M}(\alpha,\beta)=\{\alpha,\beta\} \tag{3.1}$$

where α,β are two such balanced bracketed expressions.

This description of the generative process of binary rooted trees is immediate from the identification of these trees with balanced bracketed expressions in a single variable x, such as

$\{\{x\{xx\}\}x\}\longleftrightarrow$  .

Under this identification, the binary operation of the magma takes two rooted binary trees and attaches the roots to a new common root

$(T,T')\mapsto \mathfrak{M}(T,T')=$  . (3.2)

If one takes the $\mathbb{Q}$-vector space $\mathcal{V}(\mathfrak{T})$ spanned by the set $\mathfrak{T}$, the magma operation $\mathfrak{M}$ on $\mathfrak{T}$ induces on $\mathcal{V}(\mathfrak{T})$ the structure of an algebra. More precisely, $\mathcal{V}(\mathfrak{T})$ is the *free commutative non-associative algebra generated by a single variable* x, or equivalently the *free algebra over the quadratic operad freely generated by the single commutative binary operation* $\mathfrak{M}$ (see [27]).

One can assign a grading (a weight, measuring the size) to the binary rooted trees by the number of leaves, $\ell=\#L(T)$, so that one can decompose the vector space $\mathcal{V}(\mathfrak{T})=\oplus_\ell\mathcal{V}(\mathfrak{T})_\ell$ into its graded components (the sub-vector-spaces spanned by the trees of weight ℓ).

As discussed in [37], this is a very natural and simple mathematical object, and the generative process for the binary rooted trees can then be seen as the simplest and most fundamental possible case of recursive solution of a fixed point problem, of the kind that is known in physics as Dyson–Schwinger equation.

Indeed, one can view the recursive construction of binary rooted trees through repeated application of the Merge operation (3.1) as the recursive construction of a solution to the fixed point equation

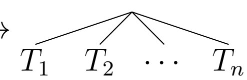
$$X=\mathfrak{M}(X,X)\,, \tag{3.3}$$

where $X = \sum_\ell X_\ell$ is a formal infinite sum of variables X_ℓ in $\mathcal{V}(\mathfrak{T})_\ell$, with ℓ the grading by number of leaves. The problem (3.3) can be solved recursively by degrees, with

$$X_n = \mathfrak{M}(X, X)_n = \sum_{j=1}^{n-1} \mathfrak{M}(X_j, X_{n-j}) \,,$$

with solution $X_1 = x$, $X_2 = \{xx\}$, $X_3 = \{x\{xx\}\} + \{\{xx\}x\} = 2\{x\{xx\}\}$, $X_4 = 2\{x\{x\{xx\}\}\} + \{\{xx\}\{xx\}\}$, and so on. The coefficients with which the trees occur in these solutions count the different possible choices of planar embeddings.

The equation (3.3) is the simplest case (with the simplest quadratic polynomial) of the more general form of Dyson–Schwinger equations for rooted trees $X = \mathfrak{B}(P(X))$ where $P(X)$ is a polynomial in the formal variable X and $\mathfrak{B}$ (usually called B^+ in the physics literature) is the operation that takes a collection of rooted trees $T_1, T_2, \ldots, T_n$, seen as the forest $F = T_1 \sqcup T_2 \sqcup \ldots \sqcup T_n$, which appears as a monomial in $P(X)$, and constructs a new tree by attaching all the roots of the T_i's to a single new root (an n-ary Merge),

(3.4) $$\mathfrak{B} : T_1 \sqcup T_2 \sqcup \ldots \sqcup T_n \mapsto$$  .

For a detailed discussion of the role of this operator in physics see, for instance, [19], [45]. The case where $P(X)$ has a single quadratic term gives the binary Merge operation in its core generative process.

3.2. **Syntactic objects.** The core computational structure of §3.1 introduces Merge in the most basic form of binary set formation (3.1). This can then be extended to the generative process that gives rise to *syntactic objects*.

One starts with an assigned set $\mathcal{SO}_0$ of *lexical items* and *syntactic features* such as $N, V, A, P, C, T, D, \ldots$

The set $\mathcal{SO}$ of syntactic objects is then identified with the set

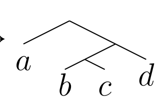
(3.5) $$\mathcal{SO} \simeq \mathfrak{T}_{\mathcal{SO}_0}$$

of finite binary rooted trees with no assigned planar embedding, and with leaves labelled by elements of $\mathcal{SO}_0$. Just as the set of binary rooted trees with no labeling of leaves has a magma structure with the binary set formation operation $\mathfrak{M}$ of (3.1), the set of syntactic objects also has a magma structure, namely it is the *free, non-associative, commutative magma over the set* $\mathcal{SO}_0$,

(3.6) $$\mathcal{SO} = \mathrm{Magma}_{na,c}(\mathcal{SO}_0, \mathfrak{M}) \,,$$

with the binary Merge operation $\mathfrak{M}$ defined as in (3.2) for pairs of trees with labelled leaves.

Note that the description as elements of the magma $\mathrm{Magma}_{na,c}(\mathcal{SO}_0, \mathfrak{M})$ is what is usually referred in the linguistics setting (see [10], [11]) as the description in terms of *sets* (in fact *multisets*) rather than *trees*, since one expresses the elements of the magma in terms of (multi)sets with brackets corresponding to the Merge operations, rather than representing them in tree form, such as

$$\{a, \{\{b, c\}, d\}\} \leftrightarrow$$  ,

where the tree on the right should not be considered as planar. As sets

$$\mathfrak{T}_{\mathcal{SO}_0} \simeq \mathrm{Magma}_{na,c}(\mathcal{SO}_0, \mathfrak{M})$$

are in bijection, hence both descriptions are equivalent, but the description as magma is preferred in linguistics as it emphasizes the generative process. The description in terms of magma elements, as in the left-hand-side of the example above, also avoids confusion as to whether the trees have a planar embedding or not: working with (multi)sets rather than with lists clearly means that no planarity is assumed. We will adopt here the tree notation, just because certain mathematical operation we will be using have a simpler and more immediately visualizable description in terms of trees rather than in terms of the corresponding multisets.

In both the core magma $(\mathfrak{T}, \mathfrak{M})$ and in the magma (3.6) of syntactic objects, one can introduce a multiplicative unit 1 satisfying $\mathfrak{M}(T, 1) = T = \mathfrak{M}(1, T)$ for all trees, by formally adding a trivial (empty) tree.

3.3. **Workspaces.** In the new formulation of Minimalism, the Merge operation acts on workspaces, consisting of material (lexical items and syntactic objects) available for computation. The Merge operation updates the workspace for the next step of structure formation. This notion of workspace is formalized in [14]. Mathematically, as discussed in [37], workspaces are just finite disjoint unions of binary rooted trees (that is, *forests*) with leaves labelled by $\mathcal{SO}_0$. Equivalently, workspaces are multisets of syntactic objects.

Thus, the set of workspaces can be identified with the set $\mathfrak{F}_{\mathcal{SO}_0}$ of *binary rooted forests with no assigned planar structure* (disjoint unions of binary rooted trees with no assigned planar structure) with leaf labels in $\mathcal{SO}_0$.

Given a workspace $F \in \mathfrak{F}_{\mathcal{SO}_0}$, the material in F that is accessible for computation consist of the lexical items and all the trees that were obtained through previous applications of Merge. This inductive definition can be rephrased more directly, by defining the *set of accessible terms* of F.

For a single binary rooted tree T with no assigned planar structure, the set $Acc(T)$ of accessible terms of T is the set of all subtrees $T_v \subset T$ given by all the descendants of a given non-root vertex of T. Indeed, these subtrees T_v are exactly all the intermediate trees obtained in an iterative construction of T starting from the lexical items or features at the leaves, by repeated application of the Merge operation (3.2). We also write $Acc'(T)$ for the set obtained by adding to $Acc(T)$ a copy of T itself, so that

$$Acc'(T) = \{T_v \,|\, v \in V_{int}(T)\} \quad \text{and} \quad Acc'(T) = \{T_v \,|\, v \in V(T)\} \,,$$

where $V_{int}(T)$ is the set of non-root vertices and $V(T)$ is the set of all vertices of T including the root.

For a workspace given by a forest $F = \sqcup_{a \in \mathcal{I}} T_a \in \mathfrak{F}_{\mathcal{SO}_0}$, with T_a the component trees and $\mathcal{I}$ a finite indexing set, the set of accessible terms *of the workspace* is given by

$$Acc(F) = \bigsqcup_{a \in \mathcal{I}} Acc'(T_a) \,, \tag{3.7}$$

namely it consists of the syntactic objects, i.e. the connected components T_a of F, together with all the accessible terms $\alpha \in Acc(T_a)$ of each component T_a.

3.4. **Merge acting on workspaces.** The action of Merge on workspaces is then defined in the new version of Minimalism as a transformation that produce a new workspace from a given one, given a choice of a pair of syntactic objects.

In $S, S' \in \mathfrak{T}_{\mathcal{SO}_0}$ are two given syntactic objects, the Merge operator $\mathfrak{M}_{S,S'}$ acts on a given workspace F in the following way. For each of the two arguments of the binary operator $\mathfrak{M}_{S,S'}$ the set $Acc(F)$ of accessible terms is searched for a term matching S, respectively S'. If matching terms are not found the workspace $F = \sqcup_a T_a$ remains the same. If they are found, say $T_{i,v_i} \simeq S$

and $T_{j,v_j} \simeq S'$, then the new resulting workspace $F' = \mathfrak{M}_{S,S'}(F)$ is of the form

$$F' = \mathfrak{M}(T_{i,v_i}, T_{j,v_j}) \sqcup T_i/T_{i,v_i} \sqcup T_j/T_{j,v_j} \sqcup \bigsqcup_{a \neq i,j} T_a \,, \tag{3.8}$$

where the quotients $T_i/T_{i,v_i}$ and $T_j/T_{j,v_j}$ perform the cancellation of the deeper copies of the accessible terms T_{i,v_i} and T_{j,v_j} in the new workspace.

The quotient T/T_v of a tree T by an accessible term $T_v \subset T$ is defined here in the following way, see [37]. One first removes the subtree T_v from T, leaving the complement $T \smallsetminus T_v$. There is then a unique maximal binary rooted tree that can be obtained from this complement $T \smallsetminus T_v$ by contracting edges, and this resulting binary rooted tree is what we call T/T_v. When v is a non-root vertex of T, the quotient T/T_v is equivalently described as obtained by contracting all of T_v to its root vertex, and also contracting the edge above the root vertex of T_v and the other edge out of the vertex above the root of T_v. In the case where $T_v = T$ we have $T/T_v = 1$. We also set $T/1 = T$.

In this new version of Minimalism, the Hopf algebra structure can be seen as underlying the action of Merge described as in (3.8). Indeed, the fundamental property of the coproduct in a Hopf algebra is providing the list of all the possible ways of decomposing an object (the term of the algebra the coproduct is applied to) into a pair of a subobject and the associated quotient object. In other words, in the case of a tree T one can write a coproduct in the form

$$\Delta(T) = \sum_{v} F_{\underline{v}} \otimes T/F_{\underline{v}} \,, \tag{3.9}$$

where we include the special cases $T \otimes 1$ and $1 \otimes T$, and where, for $\underline{v} = \{v_i\}_{i=1}^k$ we write $F_{\underline{v}}$ for the forest consisting of a union of disjoint subtrees $T_{v_i} \subset T$. This can be equivalently described as a forest obtained from an *admissible cut* of the tree T. We can view the first terms

$$\Delta_{(2)}(T) = \sum_{v} T_v \otimes T/T_v \,, \tag{3.10}$$

of the coproduct (meaning those where the subforest has a single component) as a way of compiling a list of all the accessible terms of T with the corresponding cancellation of the deeper copy. In these terms, the action (3.8) can be described as taking, for each of the two arguments of the binary operation $\mathfrak{M}_{S,S'}$ the coproduct $\Delta(F)$ over the entire workspace, which extracts all the accessible terms, searching among them for matching copies of S and S', merging them if found, and keeping the other terms of the coproduct that perform the cancellation of the deeper copies.

The $\mathbb{Q}$-vector space $\mathcal{V}(\mathfrak{F}_{\mathcal{SO}_0})$ spanned by the workspaces, with the operations of disjoint union $\sqcup$ as product and the coproduct (3.9) extended from trees to forests by $\Delta(F) = \sqcup_a \Delta(T_a)$ for $F = \sqcup_a T_a$, has the structure of an associative, commutative, coassociative, non-cocommutative bialgebra $(\mathcal{V}(\mathfrak{F}_{\mathcal{SO}_0}), \sqcup, \Delta)$. The vector space $\mathcal{V}(\mathfrak{F}_{\mathcal{SO}_0}) = \oplus_\ell \mathcal{V}(\mathfrak{F}_{\mathcal{SO}_0})_\ell$ has a grading by number of leaves, compatible with the operations, so that a coproduct making $\mathcal{V}(\mathfrak{F}_{\mathcal{SO}_0})$ a Hopf algebra can be constructed inductively by degrees.

The action of Merge on workspaces described above is formulated in [37] explicitly in terms of the product and coproduct structure of $(\mathcal{V}(\mathfrak{F}_{\mathcal{SO}_0}), \sqcup, \Delta)$, as

$$\mathfrak{M}_{S,S'} = \sqcup \circ (\mathfrak{B} \otimes \mathrm{id}) \circ \delta_{S,S'} \circ \Delta \,, \tag{3.11}$$

where the coproduct Δ produces the list of accessible terms and corresponding cancellations, and the operator $\delta_{S,S'}$ identifies the matching copies. These are then fed into Merge by the operation $\mathfrak{B} \otimes \mathrm{id}$ which at the same times produces the new merged tree through the grafting operation $\mathfrak{B}$ of (3.4) and performs the cancellation of the deeper copies by keeping the corresponding quotient terms produced by the coproduct in the new workspace. The resuling new workspace is then produced by taking all these components and the new component created by Merge together through the product operation $\sqcup$. A more detailed discussion of this operation is given in our companion paper [37].

3.5. **Internal and External Merge.** The action of Merge on workspaces defined by (3.8) and (3.11) contains other forms of Merge in addition to External and Internal Merge, such Sideward and Countercyclic Merge (see §2.3 of [37]). These are not desirable in linguistic terms, and one expects that a mechanism of Minimal Search would extract the terms corresponding to External and Internal Merge as the "least effort" contributions. Minimal Search refers to the extraction of matching terms from the list of the accessible terms produced by the coproduct, where the search in this list is done according to an appropriate economy principle.

With the formulation (3.11) of the action of Merge on workspaces, it is shown in [37] that the usual form of Minimal Search can be implemented in this Hopf algebra setting by selecting the leading order part of the coproduct with respect to a grading that weights the accessible terms $T_v \subset T$ by the distance of the vertex v from the root vertex of T, so that searching among terms deeper into the tree becomes less efficient, with respect to this cost function, than searching near the top of the tree, and taking the leading order term with respect to this grading has exactly the same effect as implementing Minimal Search in the way usually described in the linguistics literature (see [10], [11], [13]).

More precisely, one can introduce degrees in the coproduct by taking

$$\begin{gathered}\Delta^{(\epsilon,\eta)} : \mathcal{V}(\mathfrak{T}_{\mathcal{SO}_0}) \to \mathcal{V}(\mathfrak{T}_{\mathcal{SO}_0})[\epsilon] \otimes_{\mathbb{Q}} \mathcal{V}(\mathfrak{T}_{\mathcal{SO}_0})[\eta]\,, \\ \Delta^{(\epsilon,\eta)}(T) = \sum_{\underline{v}} \epsilon^{d_{\underline{v}}}\, F_{\underline{v}} \otimes \eta^{d_{\underline{v}}}(T/F_{\underline{v}})\,,\end{gathered} \tag{3.12}$$

where for $\underline{v} = \{v_1, \ldots, v_n\}$ a set of vertices $v_i \in V_{int}(T)$, we set $d_{\underline{v}} = d_{v_1} + \cdots + d_{v_n}$, with d_v the distance of a vertex v to the root of T. In the Merge action one correspondingly obtains

$$\mathfrak{M}^{\epsilon}_{S,S'} = \sqcup \circ (\mathfrak{M}^{\epsilon} \otimes \mathrm{id}) \circ \delta_{S,S'} \circ \Delta^{(\epsilon,\epsilon^{-1})} \tag{3.13}$$

with $\Delta^{(\epsilon,\epsilon^{-1})}$ as in (3.12), and with

$$\begin{gathered}\mathfrak{M}^{\epsilon} : \mathcal{V}(\mathfrak{T}_{\mathcal{SO}_0})[\epsilon,\epsilon^{-1}] \otimes_{\mathbb{Q}} \mathcal{V}(\mathfrak{T}_{\mathcal{SO}_0})[\epsilon,\epsilon^{-1}] \to \mathcal{V}(\mathfrak{T}_{\mathcal{SO}_0})[\epsilon,\epsilon^{-1}] \\ \mathfrak{M}^{\epsilon}(\epsilon^{d}\alpha, \epsilon^{\ell}\beta) = \epsilon^{|d+\ell|}\, \mathfrak{M}(\alpha,\beta)\,.\end{gathered} \tag{3.14}$$

It is then shown in [37] that taking the leading term (namely the only nonzero term in the limit $\epsilon \to 0$) of arbitrary compositions of operators $\mathfrak{M}^{\epsilon}_{S,S'}$, one finds only (compositions of) Internal and External Merge, while all the remaining forms of Merge (Sideward, Countercyclic) are of lower order and disappear in the $\epsilon \to 0$ limit.

4. Externalization and planarization

In the new formulation of Minimalism, as we discussed above, Merge happens in the free symmetric form described by the free commutative non-associative magma $\mathrm{Magma}_{na,c}(\mathcal{SO}_0, \mathfrak{M})$ of (3.6) that constructively defines syntactic objects, which are binary rooted trees with no assignment of planar structure.

In this formulation of Minimalism, the assignment of planar structure to trees happens *after* the action of Merge has taken place, in a further process of *externalization*.

This is in contrast with older versions of Minimalism (including the case of Stabler's formulation that we discussed in the previous sections), where Merge is applied directly on planar trees.

There are suggestions, such as Richard Kayne's LCA (Linear Correspondence Axiom) ([29], [30]), that propose the replacement of externalization with a more implicit (and unique) choice of planar embeddings for trees. Irrespective of the tenability of this proposal on linguistic ground, we discuss in this section some more basic difficulties in its implementation, that arise at the formal algebraic level.

First a comment about terminology: in the linguistics literature it is customary to use the term *linearization* for the choice of a linear ordering for the leaves of a binary rooted tree. Since this creates a conflict of terminology with the more common mathematical use of the word "linearization", and the choice of a linear ordering of the leaves is equivalent to the choice of a planar embedding of the tree, we will adopt here the terminology *planarization* (of trees) instead of *linearization* (of the set of leaves). Thus, we will refer to the LCA proposal as a "planarization" rather than as a "linearization algorithm" as usually described. We trust this will not cause confusion with the readers.

4.1. **Commutative and non-commutative magmas.** A first important observation is that the Merge operation can happen *before* (as in the new Minimalism) or *after* the assignment of planar structure to trees (as in the old Minimalism), but *not* both consistently. What we mean by this is the following simple mathematical observation.

Just as we consider the free commutative non-associative magma $\mathrm{Magma}_{na,c}(\mathcal{SO}_0, \mathfrak{M})$ of the new Minimalism, we can similarly consider the free non-commutative non-associative magma

$$\mathcal{SO}^{nc} := \mathrm{Magma}_{na,nc}(\mathcal{SO}_0, \mathfrak{M}^{nc})$$

over the same set $\mathcal{SO}_0$. The elements of this magma $\mathcal{SO}^{nc}$ are the *planar* binary rooted trees with leaves labelled by the set $\mathcal{SO}_0$. We write the elements of $\mathcal{SO}^{nc}$ as T^π, where T is an abstract (non-planarly embedded) binary rooted tree and π is a planar embedding of T. The non-commutative non-associative magma operation is given by

$$\mathfrak{M}^{nc}(T_1^{\pi_1}, T_2^{\pi_2}) = \overbrace{T_1^{\pi_1} \quad T_2^{\pi_2}} =: T^\pi \,,$$

where now the trees $T_1^{\pi_1}$ and $T_2^{\pi_2}$ are planar and the above tree T^π is assigned the planar embedding π where $T_1^{\pi_1}$ is to the left of $T_2^{\pi_2}$. In particular now $\mathfrak{M}^{nc}(T_1^{\pi_1}, T_2^{\pi_2}) \neq \mathfrak{M}^{nc}(T_2^{\pi_2}, T_1^{\pi_1})$, unlike the case of the commutative $\mathfrak{M}$ of $\mathcal{SO}$.

There is a morphism of magmas $\mathcal{SO}^{nc} \to \mathcal{SO}$ which simply forgets the planar structure of trees, so that $T^\pi \mapsto T$. It is well defined as a morphism of magmas since the two different planar trees $\mathfrak{M}^{nc}(T_1^{\pi_1}, T_2^{\pi_2})$ and $\mathfrak{M}^{nc}(T_2^{\pi_2}, T_1^{\pi_1})$ have the same underlying abstract (non-planar) tree $\mathfrak{M}(T_1, T_2)$.

On the other hand, there is no morphism of magmas that goes in the opposite direction, from $\mathcal{SO}$ to $\mathcal{SO}^{nc}$. Indeed, if such morphism existed, then its image would necessarily be a commutative sub-magma of $\mathcal{SO}^{nc}$, but $\mathcal{SO}^{nc}$ does not contain any nontrivial commutative sub-magma. This can be seen easily as, if a tree T^π is contained in a commutative sub-magma of $\mathcal{SO}^{nc}$, then $\mathfrak{M}^{nc}(T^\pi, T^\pi)$ also is, but $\mathfrak{M}^{nc}(\mathfrak{M}^{nc}(T^\pi, T^\pi), T^\pi)) \neq \mathfrak{M}^{nc}(T^\pi, \mathfrak{M}^{nc}(T^\pi, T^\pi))$ contradicting the fact that the sub-magma is commutative.

This fact has the immediate consequence that we stated above, namely that if a free symmetric Merge takes place *before* any assignment of planar structure to trees, then it cannot also consistently apply *after* a choice of planarization. In other words, if Σ is any choice of a section of the projection $\mathcal{SO}^{nc} \to \mathcal{SO}$ (that is, an assignment of planarization) then one *cannot have* compatible Merge operations satisfying $\mathfrak{M}^{nc}(\Sigma(T_1), \Sigma(T_2)) = \Sigma(\mathfrak{M}(T_1, T_2))$. Merge can act on abstract trees as in the new Minimalism or on planar trees as in the old Minimalism, but these two view are mutually exclusive.

Descriptions of Merge at the level of planar trees, as in the old Minimalism, involve a partially defined operations with restrictions on domains, based on some label assignments. We have discussed this explicitly in the case of Stabler's formulation but analogous conditions exist in other formulations of the old Minimalism. In view of the considerations above, this can be read as evidence of the fact that one is trying to describe at the level of planar trees a Merge that is really taking place at the underlying level of non-planar abstract trees, correcting for the incompatibility described above by restricting the domain of applicability. In particular, this means that the significant complications in the underlying algebraic structure of External and Internal Merge that we have observed in the case of Stabler's formulation, similarly apply to any other formulation that places Merge after the planar embedding of trees.

This issue does not arise within the formulation of the new Minimalism, since the planarization that happens in externalization is simply a non-canonical (meaning dependent on syntactic parameters) choice of a section Σ of the projection $\mathcal{SO}^{nc} \to \mathcal{SO}$. The only requirement on Σ is to be compatible with syntactic parameters, not to be a morphism with respect to the magma operations $\mathfrak{M}$ and $\mathfrak{M}^{nc}$, since in this model Merge only acts before externalization as $\mathfrak{M}$ (not after externalization as $\mathfrak{M}^{nc}$). In our previous paper [37] we describe externalization in the form of a correspondence. This represents the two step procedure that first choses a non-canonical section Σ and then quotients the image by eliminating those planar trees obtained in this way that are not compatible with further (language specific) syntactic constraints. Merge is not applied anywhere in this externalization process, which happens *after* the results of Merge have been computed at the level of trees without planar structure.

4.2. **Planarization versus Externalization.** Proposals such as Kayne's LCA planarization of trees, [29], [30] (see also Chapter 7 of [28] and [34] for a short summary), suggest the replacement of externalization with a different way of constructing planarization. This relies on the idea that the abstract trees are endowed with additional data (related to heads, maximal projection, and c-command relation) that permits a *canonical* choice of planar structure. We discuss two different mathematical difficulties inherent in this proposal.

First let's assume that indeed additional data on the abstract syntactic tree suffice to endow them with a unique canonical choice of linearization. (We will discuss the difficulties with this assumptions below.) In this case, we would have a map, which we call Σ^{LCA} that assigns to an abstract binary rooted tree T a corresponding, uniquely defined non-planar tree $\Sigma^{LCA}(T) = T^{\pi_{LCA}}$, with π_{LCA} the linear ordering constructed by the LCA algorithm.

By our previous observations on the morphisms of magmas, we will necessarily have in general that

$$\mathfrak{M}^{nc}(\Sigma^{LCA}(T_1), \Sigma^{LCA}(T_2)) \neq \Sigma^{LCA}(\mathfrak{M}(T_1, T_2)) \,,$$

so that Σ^{LCA} *cannot be compatible* with asymmetric Merge of planar trees.

The only way to make it compatible with Merge would be to define Merge on the image of Σ^{LCA} not as the asymmetric Merge of planar trees but as

$$\mathfrak{M}^{LCA}(\Sigma^{LCA}(T_1), \Sigma^{LCA}(T_2)) := \Sigma^{LCA}(\mathfrak{M}(T_1, T_2)) \,.$$

This, however, simply creates an isomorphic copy of the commutative magma $\mathrm{Magma}_{na,c}(\mathcal{SO}_0, \mathfrak{M})$ (by a choice of a particular representative in each equivalence class of the projection $\mathcal{SO}^{nc} \to \mathcal{SO}$). This would imply that application of the planarization Σ^{LCA} has no effect, in terms of the properties of Merge, with respect to working directly with the free symmetric Merge on abstract non-planar trees. The alternative is, as in externalization, to not require any compatibility between Σ^{LCA} and Merge: in other word, even if LCA replaces externalization, Merge is still only happening in the form of free symmetric Merge, at the level of the abstract non-planar trees, and not after planarization.

We now look more specifically at the proposals for how the planarization Σ^{LCA} should be obtained, to highlight a different kind of difficulty.

In an (abstract) binary rooted tree T, two vertices v_1, v_2 are sisters if there is a vertex v of T connected to both v_1 and v_2. A vertex v of T dominates another vertex w if v is on the unique path from the root of T to w. A vertex v in T c-commands another vertex w if neither dominates the other and the lowest vertex that dominates v also dominates w. A vertex v asymmetrically c-commands a vertex w if v c-commands w and v, w are not sisters. Asymmetric c-command defines a partial ordering of the leaves of T. In order to extend this partial ordering relation, instead of using directly the asymmetric c-command relation to define the order structure, one uses maximal projections. Namely, one requires that a leaf ℓ precedes another leaf ℓ' if and only if either ℓ asymmetrically c-commands ℓ' or a maximal projection dominating ℓ c-commands ℓ'.

Even with this extension using maximal projections, there are issues in making this a total ordering, as discussed for instance in Chapter 7 of [28] and in [34].

Since we have been discussing in the previous sections the Stabler formalism, it is easy to see this same issue in terms of the labels $>$ and $<$ assigned to the internal vertices of (planar) trees in Stabler's formulation. The difference is that in Stabler one starts with a tree that already has a planar assignment and uses heads, maximal projection, and c-command to obtain a new planar projection. In that case, as we discussed already, the labels $>$ and $<$ are assigned to the root and the non-leaf vertices of a planar tree by pointing, at each vertex v, in the direction of the branch where the head of the tree T_v resides. If this assignment of labels is well defined, then the new planar structure of the tree can be obtained simply by flipping subtrees about their root vertex every time the vertex is labelled $<$, until all the resulting vertices become labelled by $>$.

The implicit assumption that makes this possible is that every subtree T_v of a given tree T has a head, that is, a marked leaf. This assumption on heads is necessary both for the labeling by $>$ and $<$ in Stabler's formalism and for the use of maximal projections for the definition of ordering in LCA, since a maximal projection is a subtree T_v of T that is not strictly contained in any larger T_w with the same head.

In the case of the labels $>$ and $<$, the problem of what label is assigned to the new root vertex, when External or Internal Merge is performed, requires restrictions on the domain of applicability of these Merge operations, depending on conditions on the labels at the heads of the trees used in the Merge operation (which makes them partially defined operations). It also requires other further steps, such as allowing Merge results to be unlabeled during derivation and labelled at the end so that successive-cyclic raising can remove the elements responsible for un-labelability. (This refers to the same linguistic problem with Stabler's formalism that we discussed above, at the end of §2.8, mentioned to us by Riny Huijbregts).

In the case of LCA, the problem again arises from the fact that a result of Merge need not have the heads of the two merged trees in an asymmetric c-command relation. This implies that, even with the introduction of heads and maximal projections, the assumption that all trees have a head marked leaf cannot always be satisfied, hence one cannot obtain a total ordering of the leaves (a unique choice of planar embedding).

Partial corrections to this problem in LCA are suggested using movement (see [28] p.230-231), which is similar in nature to the point mentioned above regarding un-labelable structures in Stabler's formalism, or by introducing "null heads" in the structure, or by morphological reanalysis that hides certain items from LCA. In any case, the fundamental difficulty in constructing a planarization algorithm based on heads and maximal projections can be summarized as follows.

We define a *head function* on an (abstract) binary rooted tree T as a function $h_T : V^o(T) \to L(T)$ from the set $V^o(T)$ of non-leaf vertices of T to the set $L(T)$ of leaves of T, with the property that if $T_v \subseteq T_w$ and $h_T(w) \in L(T_v) \subseteq L(T_w)$, then $h_T(w) = h_T(v)$. We write $h(T)$ for the value of h_T at the root of T. This general definition is designed to abstract the properties of the head in the syntactic sense.

Consider then pairs (T, h_T) and $(T', h_{T'})$ of trees with given head functions. There are exactly two choices of a head function on the Merge $\mathfrak{M}(T, T')$, corresponding to whether $h(T)$ or $h(T')$ is equal to $h(\mathfrak{M}(T, T'))$. Since the trees T, T' are not planar and $\mathfrak{M}$ is the symmetric Merge, there is no consistent way of making one rather than the other choice of $h_{\mathfrak{M}(T,T')}$, at each application of $\mathfrak{M}$. This implies that, on a given binary rooted tree T there are $2^{\#V^o(T)}$ possible head functions. We can think of any such choice as the assignment, at each vertex $v \in V^o(T)$ of a marking to either one or the other of the two edges exiting v in the direction away from the root.

By thinking of h_T as an assignment of a marking to one of the two edges below each vertex, the head function h_T determines a planar embedding of T by putting under each vertex the marked edge to the left. Thus, the problem of constructing planar embeddings of abstract binary rooted trees can be transformed into the problem of constructing head functions.

The LCA algorithm aims at obtaining a special assignment $T \mapsto h_T$ of head functions h_T to abstract binary rooted trees T that is somehow determined uniquely by the properties of the labeling set $\mathcal{SO}_0$ of the leaves of the trees. Let us denote by $\lambda(\ell)$ the label in $\mathcal{SO}_0$ assigned to the leaf $\ell \in L(T)$.

This is where the main difficulty arises. For instance, if the labeling set $\mathcal{SO}_0$ happens to be a totally ordered set (which is not a realistic linguistic assumption), as long at two trees (T, h_T) and $(T', h_{T'})$ have head functions with labels $\lambda(h(T)) \neq \lambda(h(T'))$, there is always a preferred choice of head function on $\mathfrak{M}(T, T')$, which is the one in which the two subtrees T and T' are ordered according to the ordering of the labels $\lambda(h(T))$ and $\lambda(h(T'))$ in $\mathcal{SO}_0$. However, this excludes the case where the leaves $h(T)$ and $h(T')$ may have the same label in $\mathcal{SO}_0$. So even under the unrealistically strong assumption that labels are taken from a totally ordered set, a planarization algorithm based on the construction of a head function cannot be defined on all the syntactic objects in $\mathcal{SO}$.

At the level of the underlying algebraic structure, the issue with planarization is therefore twofold. It does not provide an alternative to Merge acting on the non-planar trees, for the reasons mentioned above on maps of magmas. At the same time, a choice of planar embedding that is independent of syntactic parameters and is based only on heads of trees would require a canonical construction of head functions from properties of the labeling set $\mathcal{SO}_0$, but this cannot be done consistently on the entire set of syntactic objects $\mathcal{SO}$ produced by the free symmetric Merge $\mathfrak{M}$.

5. Conclusions

We have seen from this comparative analysis of the algebraic structures underlying one of the older versions of Minimalism (Stabler's Computational Minimalism) and Chomsky's newer version of Minimalism that the new version has a simpler mathematical structure with a unifying description of Internal and External Merge, and with a core generative process that reflects the most fundamental magma of binary set formation, generating the binary rooted trees without planar structure.

The more complicated mathematical structure of Stabler's Minimalism is caused by several factors. The intrinsic asymmetry of the Internal Merge is due to working with *planar* binary rooted trees. Abandoning the idea that planar embeddings should be part of the core computational structure of Minimalism is justified linguistically by the relevance of structures (abstract binary rooted trees) rather than strings (linearly ordered sets of leaves, or equivalently planar embeddings of trees) in syntactic parsing, see [18]. Thus, working with abstract trees without planar embeddings is one of the simplifying factors of the new Minimalism. The other main issue that complicates the mathematical structure of the older versions of Minimalism is the very different nature of the Internal and External Merge operations: in the case of Stabler's Minimalism analyzed here these two forms of Merge relate to two very different algebraic objects (operated algebras and right-ideal coideals) hence they cannot be reconciled as coming from the same operation, while in the new Minimalism both Internal and External Merge are cases of the same operation, and both arise as the leading terms with respect to the appropriate formulation of Minimal Search. Finally, another main issue that makes the mathematical structure of older versions of Minimalism significantly more complicated is the presence of conditions on labels that need to be matched for Internal and External Merge to be applicable, related to the problem of projections discussed in [9]. Mathematically this makes all operations only partially defined on particular domains where conditions on labels are met. As we discussed, this creates problems with having to work with partially defined versions of various algebraic structures and it significantly complicated iterations of the Merge operations, where the conditions on domains compound. Since the conditions on labels are absent from the fundamental structure of the new Minimalism, this problem of dealing with partially defined operations also disappears, leading to another simplification at the level of the mathematical structures involved.

Within the new formulation of Minimalism, the planar structure of trees is introduced as a later step of externalization, not at the level of the Merge action. The choice of planar structure in externalization is done through a non-canonical (that is, dependent on syntactic parameters) section of the projection from planar to abstract trees. We show that proposed construction of a unique canonical choice of planar embeddings, based on heads of trees, maximal projections, and c-command, run into difficulties at the level of the underlying algebraic structure.

Acknowledgments. We thank Riny Huijbregts for helpful comments and feedback, and Richard Kayne for useful discussions. The first author acknowledges support from NSF grant DMS-2104330 and FQXi grants FQXi-RFP-1 804 and FQXi-RFP-CPW-2014, SVCF grant 2020-224047, and support from the Center for Evolutionary Science at Caltech.

References

[1] M. Aguiar, F. Sottile, *Structure of the Loday–Ronco Hopf algebra of trees*, Journal of Algebra, Vol.295 (2006) 473–511.
[2] F. Baader, T. Nipkow, *Term Rewriting and All That*, Cambridge University Press, 1998.
[3] R.C. Berwick, *Mind the gap*, in "50 Years Later: Reflections on Chomsky's Aspects", pp. 1–13, MITWPL, 2015.
[4] R.C. Berwick, N. Chomsky, *Why only us?* MIT Press, 2017.
[5] R.C. Berwick, S. Epstein, *On the Convergence of 'Minimalist' Syntax and Categorial Grammar*, in "Algebraic Methods in Language Processing 1995", pp. 143—148, Universiteit Twente, 1995.
[6] A. Borowiec, W.A. Dudek, S. Duplij, *Basic concepts of ternary Hopf algebras*, arXiv:0306208, Journal of Kharkov National University, ser. Nuclei, Particles and Fields, v. 529, N 3(15) (2001) pp. 21–29.
[7] Ch. Brouder, A. Frabetti, *QED Hopf algebras on planar binary trees.* J. Algebra 267 (2003) N.1, 298–322.
[8] N. Chomsky, *The Minimalist Program*, MIT Press, 1995.
[9] N. Chomsky, *Problems of projection*, Lingua, Vol.130 (2013) 33-49.
[10] N. Chomsky, *Some Puzzling Foundational Issues: The Reading Program*, Catalan Journal of Linguistics Special Issue (2019) 263–285.
[11] N. Chomsky, *The UCLA lectures*, 2019. `https://ling.auf.net/lingbuzz/005485`
[12] N. Chomsky, *Simplicity and the form of grammars*, Journal of Language Modelling, Vol.9 (2021) N.1, 5–15.

[13] N. Chomsky, *Working toward the Strong Interpretation of SMT*, 2023 Theoretical Linguistics at Keio-EMU Linguistics as Scientific Inquiry Lecture Series,
https://www.youtube.com/playlist?list=PLWXQYx-RCmeP7B2UtIA8OJsvAF-xvjDuZ
[14] C. Collins, E. Stabler, *A Formalization of Minimalist Syntax*, Syntax 19 (2016) N.1, 43–78.
[15] A. Connes, D. Kreimer, *Hopf algebras, Renormalization and Noncommutative geometry*, Comm. Math. Phys 199 (1998) 203–242
[16] A. Connes, M. Marcolli, *Noncommutative Geometry, Quantum Fields, and Motives*, Colloquium Publications, Vol.55, American Mathematical Society, 2008.
[17] C. Delaney, M. Marcolli, *Dyson-Schwinger equations in the theory of computation*, in "Feynman amplitudes, periods and motives", pp. 79–107, Contemp. Math. 648, Amer. Math. Soc., 2015.
[18] M.B.H. Everaert, M.A.C. Huybregts, N. Chomsky, R.C. Berwick, J.J. Bolhuis, *Structures, Not Strings: Linguistics as Part of the Cognitive Sciences*, Trends in Cognitive Sciences, Vol.19 (2015) N.12, 729–743.
[19] L. Foissy, *Classification of systems of Dyson–Schwinger equations in the Hopf algebra of decorated rooted trees*, Advances in Math. 224 (2010) 2094–2150.
[20] L. Foissy, *Les algèbres de Hopf des arbres enracinés, I*, Bull. Sci. Math. 126 (2002) 193–239.
[21] S. Fong, J. Ginsburg, *On constraining Free Merge*, The 43rd Meeting of the Kansai Linguistics Society. Konan University: Kobe, Japan, 2018.
[22] X. Gao, L. Guo, H. Zhang, *Rota's program on algebraic operators, rewriting systems and Gröbner-Shirshov bases*, arXiv:2108.11823.
[23] L. Gerritzen, R. Holtkamp, *Hopf co-addition for free magma algebras and the non-associative Hausdorff series*, J. of Algebra, Vol. 265 (2003) 264–284.
[24] L. Guo, *Operated semigroups, Motzkin paths and rooted trees*, J. Algebraic Combin. 29 (2009), 35–62.
[25] R. Holtkamp, *Comparison of Hopf algebras on trees*, Arch. Math. (Basel) 80 (4) (2003) 368–383.
[26] R. Holtkamp, *Rooted trees appearing in products and co-products*, in "Combinatorics and physics", 153–169, Contemp. Math., 539, Amer. Math. Soc., 2011.
[27] R. Holtkamp, *A pseudo-analyzer approach to formal group laws not of operad type*, J. Algebra, Vol. 237 (2001) 382–405.
[28] N. Hornstein, J. Nunes, K.K. Grohmann, *Understanding Minimalism*, Cambridge University Press, 2005.
[29] R.S. Kayne, *The Asymmetry of Syntax*, MIT Press, 1994.
[30] R.S. Kayne, *Temporal/Linear order, antisymmetry and externalization*, Research in Generative Grammar 45 (2023) N.2. 1–22.
[31] M. Komachi, H. Kitahara, A. Uchibori, K. Takita, *Generative procedure revisited*. Reports of the Keio Institute of Cultural and Linguistic Studies, 50 (2019) 269–283.
[32] D. Kreimer, *On the Hopf algebra structure of perturbative quantum field theories*, Adv. Theor. Math. Phys. 2 (1998) N.2, 303–334.
[33] D. Kreimer, W.D. van Suijlekom, *Recursive relations in the core Hopf algebra*, Nuclear Phys. B 820 (2009), no. 3, 682–693.
[34] N. LaCara, *The Linear Correspondence Axiom*, LIN331 Syntactic Theory, course at the University of Toronto, 26 July 2018.
[35] J.L. Loday, M. Ronco, *Hopf algebra of the planar binary trees*, Adv. Math. 139 (1998) N.2, 293–309.
[36] J.L. Loday, M. Ronco, *Order structure on the algebra of permutations and of planar binary trees*, J. Alg. Combin. Vol. 15 (2002) N.3, 253–270.
[37] M. Marcolli, N. Chomsky, R.C. Berwick, *Mathematical structure of syntactic Merge*, preprint 2023.
[38] A. Masuoka, *Quotient Theory of Hopf algebras*, in (J. Bergen, S. Montgomery, Eds.), "Advances in Hopf Algebras", Marcel Dekker, 1994, pp. 107–133.
[39] J. Michaelis, *Transforming linear context free rewriting systems into minimalist grammars*, in "Logical Aspects of Computational Linguistics (NY, 2001)" (P. de Groote, G. Morrill, and C. Retoré, Eds.), Lecture Notes in Artificial Intelligence, Vol. 2099, Springer, pp. 228–244.
[40] G.C. Rota, *Baxter operators, an introduction*, in (Joseph P.S.Kung, Ed.), "Gian-Carlo Rota on Combinatorics, Introductory papers and commentaries", Birkhäuser, 1995, pp. 504–512.
[41] P. Schauenburg, H.J. Schneider, *On generalized Hopf Galois extensions*, Journal of Pure and Applied Algebra, Vol.202 (2005) 168–194.
[42] E.P. Stabler, *Computational perspectives on minimalism*, in "Oxford Handbook of Linguistic Minimalism" (C. Boeckx, ed.), Oxford University Press, 2010, 616–641.
[43] M. Takeuchi, *Quotient spaces for Hopf algebras*, Comm. Algebra 22 (1994), N.7, 2503–2523.

[44] K. Vijay-Shanker, D. Weir, *The equivalence of four extensions of context free grammar formalisms*, Mathematical Systems Theory, 27 (1994) 511–545.
[45] K. Yeats, *Rearranging Dyson-Schwinger Equations*, Memoirs of the American Mathematical Society, 211, American Mathematical Society, 2011.
[46] Y. Zhang, X. Gao, *Hopf algebras of planar binary trees: an operated algebra approach*, Journal of Algebraic Combinatorics, 51 (2020) 567–588.

Departments of Mathematics and of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125, USA
Email address: `matilde@caltech.edu`

Institute for Data, Systems, and Society, Massachusetts Institute of Technology, Cambridge MA 02141, USA
Email address: `berwick@csail.mit.edu`

Department of Linguistics, University of Arizona, Tucson, AZ 85721, USA
Email address: `noamchomsky@email.arizona.edu`
Email address: `chomsky@mit.edu`