Chapter 5

Glue Semantics for Hopf-Algebraic Minimalism

Avery Andrews[a]
[a]Australian National University

Recent works by Marcolli, Chomsky and Berwick have developed the observation that the Hopf algebra of trees, originally developed to solve a problem in physics, has a surprisingly good fit to the basic ideas of the most recent version of Minimalist syntax. They have also proposed a number of ideas about how to do semantics, but these seem to depend on an incompletely worked out utilization of traces. Here I will suggest a way of using the glue semantics of Lexical-Functional Grammar to do semantic interpretation on Hopf-algebraic syntactic derivation without traces, accommodating the basics of quantifier scope without syntactic movement. A diagrammatic notation based on Construction Grammar is used for the glue proofs.

The purpose of this paper is to use the 'glue semantics' of Lexical-Functional Grammar (LFG) to provide a form of semantic interpretation for the new formulation of some of the basics of Minimalist syntax using Hopf algebras presented in Marcolli, Berwick, et al. (2023), Marcolli, Chomsky, et al. (2023a) and several lectures and lecture series, here collectively referred to as MCB.[1] Glue semantics is essentially Montague-style semantics attached to a subset of Girard's Linear Logic in such a way that it can do compositional semantics on structures that are not trees, such as LFG's f-structures, which fail to be trees due to containing reentrancies, and can furthermore induce scope relations on structures (such as, originally, LFG's f-structures) that do not have enough hierarchical structure to directly support them.

The Hopf-algebraic version of Minimalism arises from the observation by Marcolli that the Hopf algebra of unordered trees, originally developed for dealing

[1]These include the lecture Marcolli (2023a), the lecture series Marcolli (2023b) and the course materials Marcolli (2024), especially Marcolli et al. (2024).

Avery Andrews. 2025. Glue Semantics for Hopf-Algebraic Minimalism. In Aikaterini-Lida Kalouli, Tracy Holloway King, Stephen Pulman & Annie Zaenen (eds.), *Semantics at the Crossroads: From Theoretical Explorations to Implementations*, 79–108. Konstanz: PubliKon.

with renormalization in Feynmann diagrams, provides the basic machinery for External and Internal Merge as proposed in the 'New Minimalism' as developed in syntax since 2013, recently presented in Chomsky et al. (2023), while not fitting so well with older versions, such as the one formalized by Stabler (2010). This apparent compatibility with the most recent version of Minimalism might of course be an accident, with no real significance for linguistics, but I think that it is nevertheless sufficiently intriguing to be worth looking at carefully.

The ingredients of this new mathematical formulation are:

(1) a. The Hopf algebra of unordered trees, presented concisely by Foissy (2013).[2] This provides a concept of 'workspace' as essentially a set or multiset of trees,[3] together with a 'coproduct' that dissects trees into an extracted part and a remainder. The coproduct provides the materials for Internal Merge, with removal of the extracted material from its original position. There is also an operation of disjoint union for workspaces, which plays a role in the formulation of the syntactic Merge operations.

b. The B^+ operator, which combines a subset of trees in a workspace into a single tree. This does the tree assembly for syntactic Merge.

c. Additional concepts to manage linearization and other parameters, that are not our concern here.

Appendix B provides a brief discussion to the basics of Hopf algebras including of trees, with further references.

Some features of New Minimalism that are covered by this formulation are:

(2) a. Trees are (mostly) binary, are unordered, and only terminals have properties apart from the contents of their daughters.

[2]Foissy does not restrict the trees to binary, while for the most part, MCB want to do that, but allow non-binary trees in the process of 'Form Set' (Marcolli et al. 2024: 137-142). They argue that this is not actually Merge, but the argument is incomplete, because, for example we can presumably combine a trinary coordinate structure and a binary to produce a binary coordinate structure (*Tom, Mary and Alice, and Bill and Susan*, perhaps a guest list comprising a polycule and a couple). Whether the Hopf algebra itself should be restricted to binary trees only, or the evident restriction should be explained in terms of something else, such as Theta role assignment, deserves further thought.

[3]Which it is depends on aspects of formalization; if trees are taken to be sets, then the workspace must be a multiset, since it must be able to contain copies, but if they are graphs, then it can be a set. The latter formulation seems to be the most popular, and will be assumed here, but we will also find a use for sets.

b. Labels are found by searching for material on terminals, not on nonterminals.

c. Merge is restricted to External and Internal, some other possibilities, such as sideways and countercyclic are excluded. (This is achieved by adding some parameters to the formulation, and taking them to 0 as a limit.)

This degree of convergence is rather intriguing, and makes the Hopf-algebraic formulation worth looking into.

Semantics has been considered in this framework in Marcolli, Chomsky, et al. (2023b) and other sources, including Marcolli et al. (2024), Marcolli (2024) and sources cited there. The latter two include a discussion of how to include conventional Montague-based formal semantics in the approach, adapting the Heim & Kratzer (1998) formulation of the syntax-semantics interface. However these accounts do not yet appear to provide an account of scope, either of elements that take wide scope *in situ*, such as quantifiers in English, nor those that get displaced to become adjoined to their scope domain, such as *wh*-words in English. However in Dalrymple et al. (1997), it was shown that 'glue semantics' can assign the correct scope interpretations in some complicated cases where classic 'Quantifier Raising' fails. Furthermore, Andrews (2007, 2008, 2019) argues that glue semantics can be adapted to give reasonably clean accounts of idiomatic/multiword expressions, as well as problems of semantically idiosyncratic government of cases and adpositions, another range of problems not considered in the currently envisioned treatments of semantics in Hopf-algebraic Minimalism.

For these reasons, I think it is worthwhile to see whether glue semantics can be applied to the Hopf-algebraic version of Mininalism.[4] The version I will use is the 'propositional glue' of Andrews (2010), which dispenses with the use of linear universal quantification as standardly used in the treatment of scope (in the logic of semantic assembly in glue, not the NL semantics version of universal quantification), instead using the LFG version of 'instantiation', which we will come to see later. Gotham (2021) argues that propositional glue does not give as good a treatment of certain scope restrictions as the scope restrictions as the version with universal quantification, but propositional glue is a bit simpler than either first order glue (Kokkonidis 2008), or System F as used in the now standard 'new glue' of Dalrymple et al. (1999), and also constitutes a free symmetric monoidal closed category (Troelstra 1992: 81-92).

[4]See Gotham (2018) for its application to an older version of Minimalism.

Another feature of the approach presented here is that the semantics is applied incrementally, as the grammatical structure is produced by Merge, rather than to completed syntactic outputs as presented in Marcolli et al. (2024) and other sources cited above. This avoids the use of traces, whose functioning in semantics does not seem to be fully worked out, and it is in any event interesting to see whether interpretation can be done at all without them. A further difference is that in this version, we make no use of the concept of 'head' and therefore of 'head functions' defined over the syntax. These of course might start being used as more syntactic phenomena are considered.

I will finally introduce the presentational, but not substantive, innovation of presenting the glue proofs in a box notation inspired by that of construction grammar (CxG) as used in Kay & Fillmore (1999), although enhanced with a means to directly represent quantifier scope. The motivation for this is that many people find the deductive presentation of glue intimidating, and perhaps the CxG-inspired notation will make it easier. Another is that I think this partial resemblance to CxG is inherently interesting. For mainstream introductions to glue, see Dalrymple (2001), Asudeh (2022) and Asudeh (2023).

It is a feature of glue that semantic assembly can proceed autonomously, without constraints from syntax. I will therefore start by introducing glue and the box notation in that manner, and then show how to link glue assembly to the Hopf-algebraic syntax.

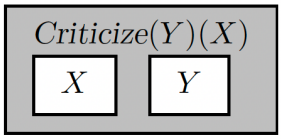
1 Glue by itself

1.1 Applying predicates to arguments

We begin with the apparently simplest semantic contributions from the point of view of compositional semantics, proper names, which, following CxG notation, we will represent as boxes containing a label for their meaning (whose connection to pragmatics is well-known to be extremely complicated). However we differ from standard CxG notation in having the boxes be shaded:

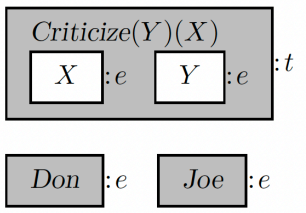
(3) 

For the next-most-complex kind of contribution, predicates with arguments, in addition to the shaded outer box, there are unshaded inner boxes, which contain upper case variables, and the meaning label repeats these variables as arguments to the predicate:

(4) 

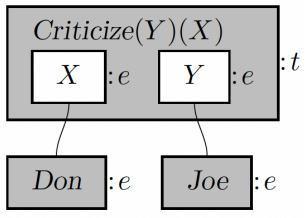
The shading represents 'polarity' in (intuitionistic implicational) proof-nets, as explained by de Groote (1999), with the difference from de Groote that we have taken the shaded boxes to be positive, representing the 'presence of content' to be sent elsewhere, the unshaded ones to be negative, representing an absence of content, to be remedied by the importation of content from a shaded, positive box. There is a further fundamental issue of whether the order of applications, as exists by necessity if curried functions as employed by Montague are used, has any linguistic meaning; I tend to take arguments from the 1970s and early 1980s, summarized and extended by Marantz (1984), as indicating that it does, with the less active arguments composed earlier (with a considerable range of variation, as discussed in the literature on Icelandic starting with Bernódusson (1982), and numerous discussions of double object constructions in many frameworks). Because of our use of upper case substitution variables, we can arrange the boxes so as to reduce crossing of the connecting lines, with their linear ordering having no significance, but we could also convert to an order-based notation (discussed in appendix A, and better if you want to write in semantic values with the substitutions applied).

The next elaboration over CxG is that each box has a type, which are the types as usual in basic formal semantics, using simple type theory. Conventionally, people start with types *e* for 'entity', and *t* for things at least somewhat like propositions (with truth values), and so shall we, at the beginning; after introducing them, people tend to omit them in practice. When we write them, we will write them to the right of the boxes, separated by colons (the types of all boxes are atomic, as discussed in the final section). So now the basic semantic ingredients for a sentence such as *Don criticized Joe* can be represented as:

(5) 

It should now be clear that the technique of semantic assembly is going to be to connect shaded boxes to unshaded ones, matching the types. This gives us two ways of hooking up the boxes in (5), one of which is depicted below:

(6)



The links are called 'axiom links', because they represent the use of the (sole) axiom $a \vdash a$ in the Gentzen sequent calculus, the basis for proof-nets.

In fact, it appears that Kay and Fillmore came up with the basic idea of proof-nets for the easier half of intuitionistic implicational logic, but did not to my knowledge finish the formalization, which we do by presenting the rules governing legitimate hookups with axiom links (which allow all and only the valid sequent calculus deductions using the 'implication left' rule). To formulate the rules, we want another concept, the 'dynamic graph', introduced by Lamarche (1994), and used by de Groote (1999), with the directionality reversed. We follow de Groote, and, for us, the dynamic graph is composed of two kinds of oriented links:

(7) a. Axiom links, going from shaded to unshaded boxes.

b. Implicit links, going from unshaded boxes to their immediately containing shaded box (these links are explicit in the usual proof-net notation).

The rules are then as follows:

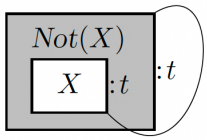
(8) a. Each unshaded box must be axiom-linked to one and only one shaded box.

b. Each shaded box is axiom-liked to at most one unshaded box.

c. The dynamic graph forms a rooted (non-planar) tree.

The first two conditions seem to be implicitly obeyed in CxG, while the third rules out combinations such as:

(9a) 

(9b) 

There are other ways in which these configurations could be excluded, but (8c) is the choice that will be needed later.

1.2 Forming lambda-abstracts

Montague analysed quantifiers such as *everyone* (including what linguists would be likely to call 'quantified noun phrases', such as *every sensible person*) as being of the type $<<e,t>,t>$, that is, predicates that apply to a complex argument of type $<e,t>$ (e.g. *hops*) to produce a truth-value. A problem with this is that such quantifiers also appear in object position, sometimes in combinations such as *somebody loves everybody*, which are perceived as ambiguous. This problem is normally solved by some kind of rule of 'Quantifier Raising'. Glue uses something that can be regarded as yet another version of this, but implemented in the semantics rather than in the syntax.[5]

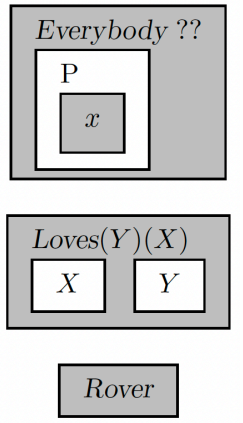
In the present adaptation of the CxG box notation, the essential idea is to allow unshaded boxes to themselves contain shaded boxes. The semantic value of these 'inner shaded boxes' is a lower-case variable, that will wind up being bound by a lambda in the semantic formula.[6]

A preliminary version of the constructors for the meaning of *Rover loves everybody* would be (types omitted because in glue they usually are, when they can be understood from context):

[5]Discussed in greater detail for earlier Minimalism by Gotham (2018).

[6]These are essentially 'inner values' in the the algebraic correctness criterion of de Groote (1999), while the semantic labels without their arguments are 'outer values'.

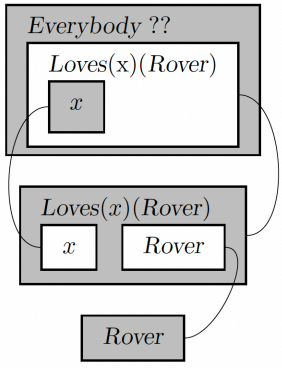
(10)



It is clear that to get the intended meaning, we will want to axiom-link the Rover box to the X box. Then to get a lambda abstract on x, we will want to link the inner shaded x box to the unshaded Y box. But what next?

Part of the answer is that first we connect the shaded *Loves* box to the unshaded P box, which we would expect to result in P being evaluated to $Loves(x)(Rover)$:

(11)



But what then do we put in for the '??' in the *Everybody* box? The answer is that we put the value of the argument-box bound by the variable in its inner shaded box, so that the pre-substitution formula for '??' is $\lambda x.P$, and the result of substitution is:

(12) $\lambda x.Loves(x)(Rover)$

The general rule is then that if an unshaded box contains a shaded box, then, in the argument position for the unshaded box contents, one writes the unshaded

box's upper case (naive substitution) variable, lambda-bound by the unshaded box's lower case variable (genuine lambda-calculus variable, not manageable by naive substitution). If the unshaded box contains more than one shaded box, then the content of the unshaded box must be specified as bound by all of the variables of the shaded boxes.

There is then a final clause to the rules governing axiom linking, which assures that this always makes sense:

(8) d. The dynamic path for an inner shaded box must pass through its immediately containing unshaded box.

Note in particular that there is *no* direct dynamic path link from an inner shaded box to its immediately containing unshaded box; the path between them must consist of other links.

1.3 A bit more on types

Before moving on to the connection to syntax, it is useful to say a bit more about types, in particular, the relationship between semantic types and glue types. In linear logic by itself, atomic propositions belong to ('inhabit') basic types. In formal semantics, the basic types correspond to ontological categories such as 'entity' and 'proposition', and people usually want to have more than one of them, although the need for this is not entirely beyond question.[7] From the basic types, we form derived types, corresponding to non-atomic propositions.

However in glue, linear logic is not by itself, but functioning as part of an interface to some kind of typed lambda calculus, possibly with extensions. This brings up the question of the relationship between the glue types and the semantic types. So far, I am aware of three possibilities:

(13) a. There is only one basic glue type, but semantic composition is regulated by composition of the semantic types.

b. The glue types are in one-to-one correspondence with the semantic types. This would include the possibility that in semantics also, there is only one basic type.

c. The two systems are different, requiring a well-formed assembly to satisfy two different sets of type constraints at the same time.

[7] See for example Partee (2009) and Partee & Borschev (2004).

The first two options are identical in effect, as far as I can determine, so it does not matter which one we choose. The third is potentially different, for example we might have a basic glue type 'predicate' corresponding to the semantic type $<e, t>$. However although this idea seems intriguing, I have not managed to derive any useful results from it. Therefore I will assume either one of the first two.

I conclude this subsection by noting that if we can retrieve the meanings of the words and idiomatic expressions in an utterance, we have a reasonable chance of retrieving the associated semantic contributions, and, especially with shorter utterances, might be able to work on the meaning without resorting to information from the syntax. However syntax is essential to comprehension in many situations, so in the next section, we show how it can be used to constrain semantic assembly, within the framework of Hopf-algebraic Minimalism.

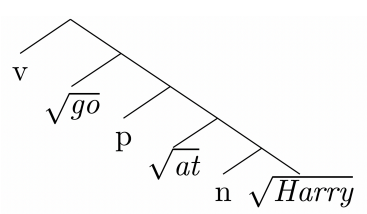
2 Semantic assembly and External Merge

The basic idea here is that of Andrews (2007, 2008, 2019), to use configurations of syntactic items to license the introduction of semantic contributions. In these papers, this is accomplished by means of a 'Semantic Lexicon', consisting of Semantic Lexicon Entries (SLEs), which specify the interpretations of combinations of conventional lexemes and features, independently of a 'Morphological Lexicon' (which probably should have just been called 'the morphology'), which specifies the morphological realization of combinations of the values of PRED ('lexical items') and inflectional features. Minimalism would use different mechanisms for packing together the features and lexical items to be realized in the morphology, which I will not discuss here. However for an example, the finite past form of *go* is *went*, across all of the idioms it appears in, such as *go off (at)* (with at least two meanings if there is no preposition) as well as in the single-word lexeme usage. I will illustrate the approach with the example *Meghan went off at Harry*, which contains some relevant syntactic problems in addition to an idiom/multiword expression (MWE).

Another respect in which this approach diverges from the Hopf-algebraic presentations so far (which do not provide fully specified trees with enough details to do syntax) is that we explicitly use categorizer formatives, such as n and v, as in various Minimalist works such as Harley (2014). This is helpful in formulating the SLEs, and also provides a clear distinction between 'lexical items' and 'features': the former co-occur with categorizers and are 'roots'; the latter do not require categorizers, and what is involved in their distribution will not be considered here.

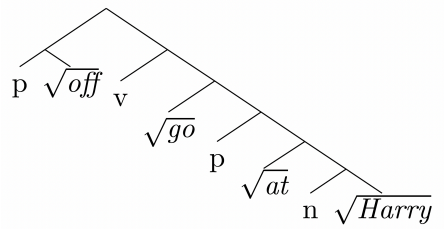
The first question is what to do about direct complements of verbs and prepositions; I will follow Harley (2014) in putting them as complements to the verb or preposition root under the appropriate categorizer, giving us (14) as the bottommost portion of the structure:[8]

(14)



Our next problem is what do do about the particle *off*, and my proposal is to place it as a 'specifier' to the v, where indirect objects might also go:

(15)



Next, we have the problem of what to do about the 'external argument' *Meghan*. I am going to defer this, and first indicate how the internal argument *Harry* and various items (here, roots) that are responsible for the meaning of the predicate are handled.

Andrews (2007, 2008, 2019) proposes that, in LFG, combinations of semantically interpretable attributes in f-structure trigger the introduction of a meaning contribution, and when they do that, they get 'checked off', so that they cannot do it again (either individually or in another combination). Furthermore, every so-interpretable attribute must get checked off (this sense of 'interpretable' is not necessarily the same as in Minimalism). Andrews (2019) proposes to formalize this by having interpretable feature instances in an f-structure contain a pointer, which must be set to a meaning-contribution, or a list of these.[9] These pointers go from interpretable feature-values to meaning-contributions. The SLEs also

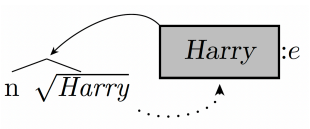
[8] Adopting the proposal of Jackendoff (1973) that particles are intransitive prepositions.

[9] Dalrymple (2001) discusses cases where it is useful to allow one item to introduce more than one meaning-contribution.

specify what I call 'semantic assembly points' in the syntactic structure, which allow the syntactic structure to constrain semantic assembly in a manner we will describe shortly.

The simplest form of SLE would be one for *Harry*:

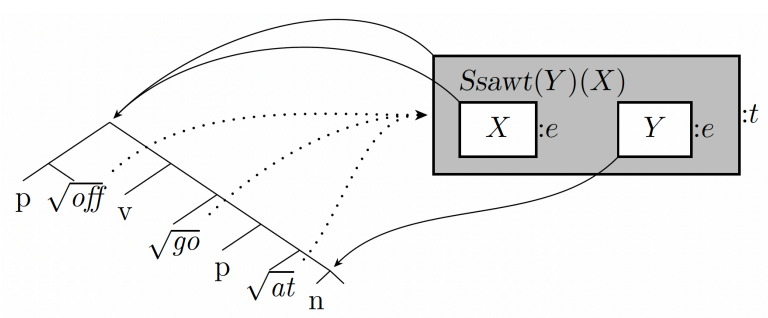
(16)



Here we take the category features to be uninterpretable, which allows their presence to be referenced by more than one SLE, which provides a straightforward way to specify the syntactic categories of the complements to a root. The dotted line from the root $\sqrt{Harry}$ represents the 'checking off' link that will cause this instance of this root in the structure to be linked to this instance of a semantic contribution, while the solid links with arrowheads are to the semantic assembly points, which function in a way to be explained later. The arrowheads represent the fact that these links are many-to-one, from the boxes in the meaning-contributions to the positions in the syntactic structure, which is important later.

The entry for the MWE *go off at* is more complex. I take its meaning to be something along the lines of *X suddenly starts saying angry words to Y*, which I will abbreviate as *Ssawt*, and it will need pointers to three semantic anchoring points, for the *at*-object, the external argument, and the resultant meaning. For reasons to be explained shortly, it works out for the second two to point to the top of the tree, yielding the following SLE:

(17)

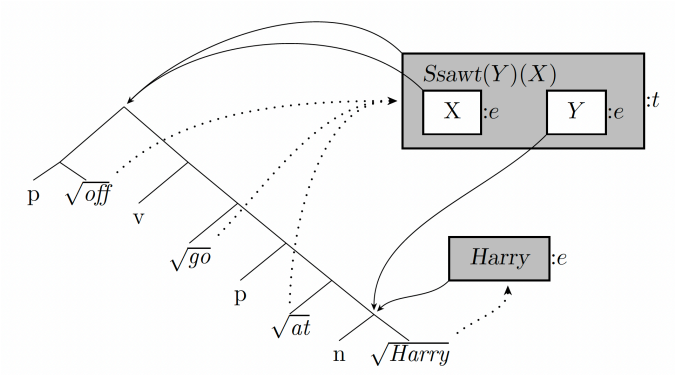


The tree on the left should be seen as a convenient and easy to read representation for a description of a tree in a tree-description language, similar in conception but different in details from the f-structure description language used in LFG. The

dotted line semantic instantiation links are redundant in the formulation of the SLE, because they will be supplied by convention to each interpretable feature in the portion of the tree that matches the description, pointing to the outermost box of the semantic contribution,[10] but are important in the structure produced. Note how the 'n' at the bottom of (17) can be the the same as that in (16), allowing the *Harry* SLE to provide a meaning for the complement of (17).

Next, consider what happens when we apply SLE's to the full tree (15):

(18)

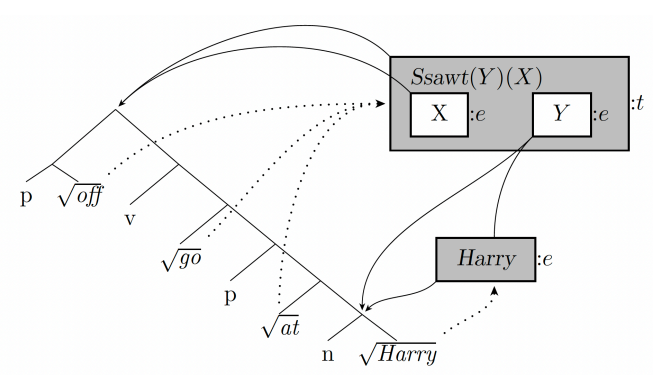


Given the SLEs introduced so far, this is the only way to attach SLEs so as to satisfy the requirement that all interpretable features get interpreted, so hooking them up with axiom links would be the next step.

The effect we want to produce is that we can axiom-link the *Harry* box to the *Y* inner box of the *Ssawt* meaning contribution, but not to its *X*-box. Visually, we can achieve this by imposing the requirement that in order to be axiom-linked, boxes must be connected by a solid arrow to the same semantic assembly point in the syntactic structure. Then this hookup is allowed, but the other conceivable one is not:

[10] In terms of function, an SLE such as (17) does essentially the same thing as a 'meaning constructor' in mainstream LFG glue, but the way it does it is different: it is matched against a generated structure ('description by analysis') as opposed to being included in a conventional lexical entry ('codescription'), as discussed by Halvorsen & Kaplan (1988).

(19)



This gives us a visual story, which we can formulate as the following principle:

(20) Axiom Link Principle: in order to add an axiom link between a shaded and unshaded box, both must be associated to the same semantic assembly point.

In terms of the deductive formulation of (propositional) glue, we want the assembly points to represent atomic propositions. This is straightforward in LFG, since these locations have always been thought of as having unique labels, but the environment here is a bit different, so we need to say more about how this can work.

The problem is that a derivation starts with a workspace consisting of a collection of atomic items (lexical items and grammatical features), which is the first in a sequence of workspaces, each created from the preceding one by an operation of External or Internal Merge, the last consisting of a single tree. Hopf algebras do not include any concept of persistent identity between nodes in successive stages of a derivation, and we do not want to add any such thing in any unnecessary way.[11] However we arguably need something of this nature for the terminals, since different instances of the same feature in a structure can have different meanings, e.g. one semantic plural and one *pluralia tantum* in *The investigators found some bathing trunks lying on the beach.* Therefore I will take from LFG the technique of assigning to each terminal, or perhaps only each interpretable terminal, a unique 'instantiation index' when it is taken from the lexicon and put in the first workspace of the derivation.

[11] And whatever was attempted would have to be consistent with the fact that the multiplication for the Hopf algebra of trees is disjoint union ($\sqcup$ in MCB), so that for a workspace W other than the empty one, it cannot be the case that $W \sqcup W = W$.

These are then packed into binary trees by a External Merge operation that combines the workspace items into trees, recursively, and Internal Merge, which uses the Hopf Algebra operation called the 'coproduct' to extract some members from the extant trees, and then further applications of the Merge operation to implement extraction.[12] These operations are explicitly formulated in such a way as to preclude anything like a 'corresponding nodes' relationship between nodes at one stage of the derivation and those of another, such as that proposed by Lakoff (1970) for Transformational Grammar.

However the instantiation indexes produced above make the terminals (or at least the interpreted ones) different, because these indexes have persistent identity through the derivation, and we can use them as 'anchors' to locate semantic assembly points via paths through the tree. If we suppose that the semantic assembly points are restricted to be non-terminals, we can specify them uniquely by means of upward paths from terminals and downward paths through nodes that have only one non-terminal daughter (recall that since the trees are unordered, there can be no specification of 'right' vs. 'left' daughter). A consequence is, for example, that if trees include *n*-ary branching nodes representing coordinate structure, an SLE will not be able to identify any of these daughters via a downward path, which would seem to be a good result.

Therefore, we can specify a semantic attachment point for a box with the index of a terminal that is connected to the meaning-contribution followed by a sequence of up or down arrows that uniquely specifies a relative position to that terminal. However once an axiom link between two boxes is created, it does not matter if the shared attachment point is removed. This corresponds to the fact that when one applies implication elimination ($A, A \multimap B \Rightarrow B$) in a proof, both the minor premise and the matching term in the other premise disappear. For formulations more general than propositional glue, currently either System F or first order, we can use the instantiation indexes combined with path specifications to provide the atomic symbols for the formulas.

There is however an issue: for an SLE comprising multiple interpreted items, which one(s) should we use as the starting point of the semantic assembly path(s)? There are several possible answers, and for the purposes of this paper, it does not matter which one is chosen. Some possibilities are:

(21) a. For each attachment point, the anchor that makes the path as short as possible.

[12] There is also 'Form Set', in Marcolli et al. (2024), which combines *n* workspace items into an *n*-ary tree, but this does not yet seem to be well-integrated into the system. For an account of the semantics of coordination in glue, see Asudeh & Crouch (2002a,b).

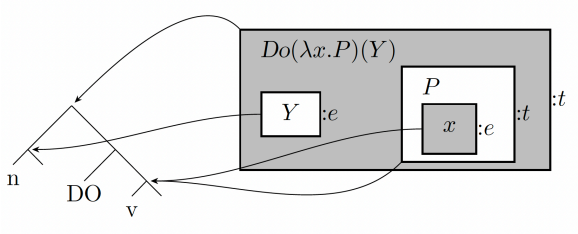
b. Free choice

c. On some basis, one of the interpreted terminals is determined as the 'prime terminal'.

(21a) is in some sense the immediately simplest, (b) seems somewhat unsatisfactory, while in order to be satisfactory, (c) would require more work to justify criteria for identifying the prime terminal. However for present purposes, any of them would suffice.[13]

Returning to more concrete grammatical issues, we need a proposal for external arguments. The approach we will take here will be similar to the one in Wood & Marantz (2017) and much other Minimalist work, in which external arguments and a range of other items, such as applicatives, are introduced by an additional item adjoined to the verb phrase. In that paper, this item is a multifunctional element i^*, but since here we are considering only external arguments, we will call it *DO*. This will take the external argument as its specifier, and the verbal phrase as its complement. Its SLE looks like this:

(22)

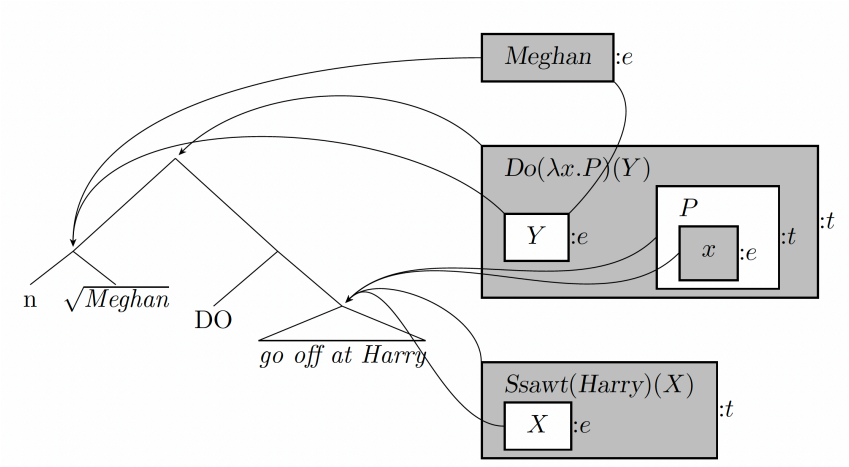


What substantive contribution *DO* makes to the meaning is a topic I will not look into here. (External arguments clearly have a strong association with agentivity, but there is likely more to it than that.)

The tree including *DO* and the external argument will then be (22) below, with the trivial axiom link for *Meghan* added, and the previously assembled structure presented as a block:

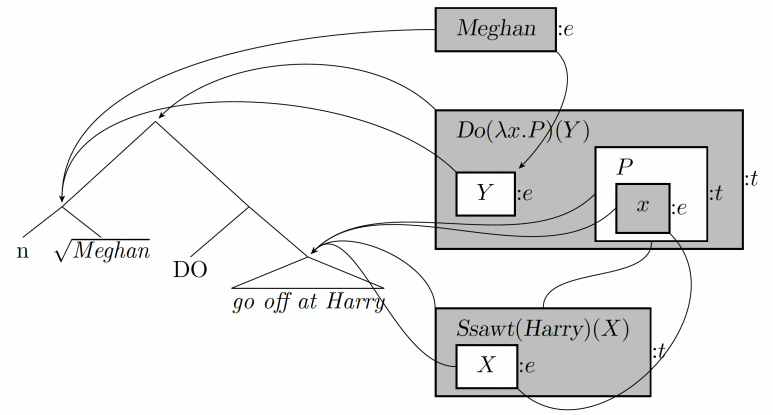
[13]There would also be the possibility of using the tree to construct a set in the obvious way (just like a decoration of a graph in the development of non-well founded set theory), with the instantiation indexes as atomic elements. However the timing of interpretation gets more critical, since if we extract something from a subtree, its associated set will change. Furthermore, if Σ is the set of sets for the trees in a workspace W, then it will also be the set of trees for the product of W by itself ($W \sqcup W$). This is not a problem for syntax, since the workspaces combined in the course of a derivation will always correspond to sets that are disjoint in the first place, since terminals are not copied.

(23)



So what about the *P* box and the predicate material? Here we have four arrows converging on one node. However two of them are of one type, and two of another, and, for each type, one comes from a shaded and the other from an unshaded box, so we can add 2 more links:

(24)



with the result that *P* gets evaluated as *Ssawt*(*Harry*)(*x*), as an argument of *DO* it gets lambda-bound by *x*, and the *DO* meaning can then combine this with *Meghan* to a final result:

(25) *DO*(λ*x*.*Ssawt*(*Harry*)(*x*))(*Meghan*).

For the reason explained above, the semantic assembly will persist if changes are made to the internal structure of (25) by Internal Merge extracting things

from it, but there are still some questions about the timing of the introduction of meaning contributions and semantic assembly.

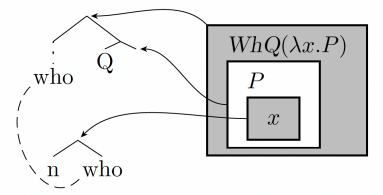
The most obvious interpretation of Andrews' previous proposals is that an SLE is attached at some point after its syntactic description is satisfied. However it does not matter when; with the structure in (15), we could attach the *Harry* SLE either as soon as the object noun phrase assembled or at some later point. We will also need the attachment of SLEs to be optional, so that lexical items can get either a meaning determined by their local context or one that they acquire later. An acceptable structure will be one that has all of its interpretable elements interpreted, but the order of operations by which this is accomplished is taken to be irrelevant.

3 Internal Merge

We now consider Internal Merge (IM), using the example of constituent *Wh*-questions, where a questioned phrase appears to move to a position where it can take its scope as a complement. For this section, it is not necessary to use an MWE as the main predicate of the example sentence; a simple transitive verb is sufficient, so that is what we will use. The complexities of external arguments are also not relevant, so we will use *Who does Rover love*, with no representation of an external argument.

Our problem is how to get *who* installed both as the object (presumably lower) argument of *like*, and how to get it to move to the position of its scope. I will suppose that *wh*-questions have an operator *Q* in complementizer position, with the clause as complement and the *Wh*-phrase as specifier. The syntactic portion of the SLE will also consist of two distinct parts, with no relations between them, but sharing a single lexical item:

(26)

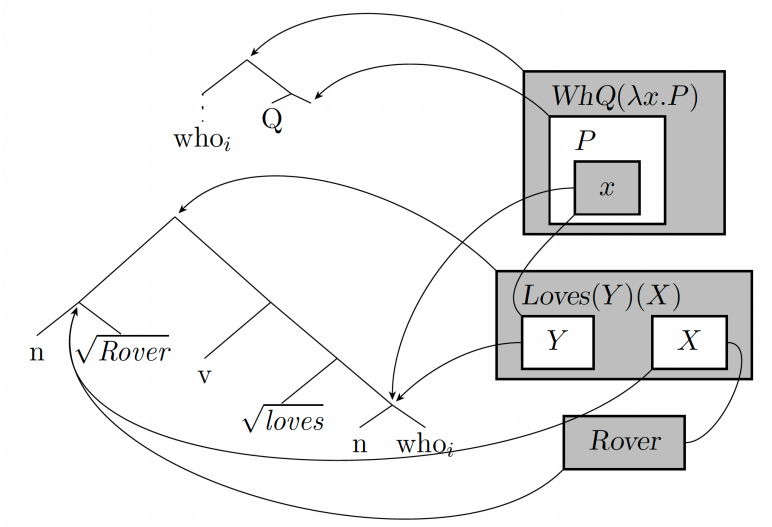


The dashed line connecting the two appearances of *who* is to indicate that they are the same syntactic object, including instantiation index, while the dotted line connecting the upper appearance to the tree-node above it indicates that it can

appear inside a containing phrase (pied piping). The nature of the constraints on this will not be considered here.

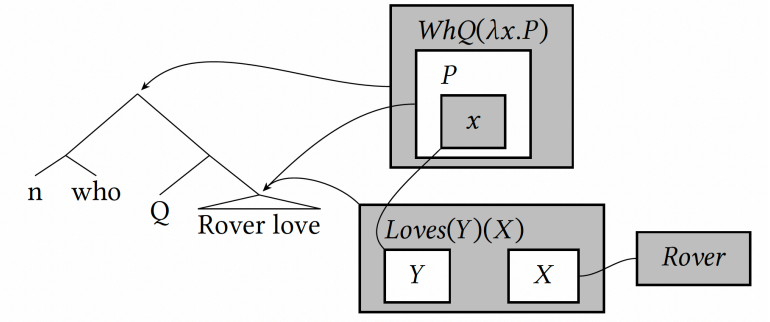
There is also an unshared item, *Q*. The proposal is that if two syntactic specifications are not connected by any tree-internal predicates, such as for example that something in one be the mother of something in the other, then they do not both have to be satisfied in the same workspace, but rather, possibly, in different ones, one derived from the other by External and Internal Merge operations. We can represent the pre-IM situation as follows, where the appearances of *who* are co-subscripted, to indicate that they are have the same instantiation index and are therefore the same item:

(27)



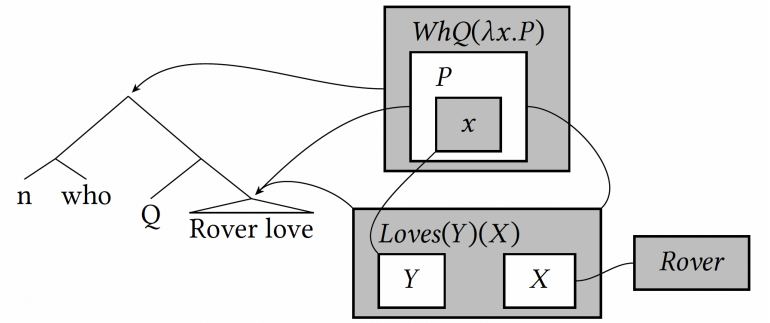
In order for the derivation to proceed, we have to Merge *Q* with the structure of (27) or with some larger one including (27) (to produce something like *Who do you think Rover loves?*). However once this is done, we can use IM to extract the questioned NP, at which point the rest of the SLE can apply, satisfying the positional requirements for the relative pronoun and providing an interpretation for the *Q* feature, and yielding:

(28)



So now we can add the final axiom link to get a well-formed interpretation:

(29)

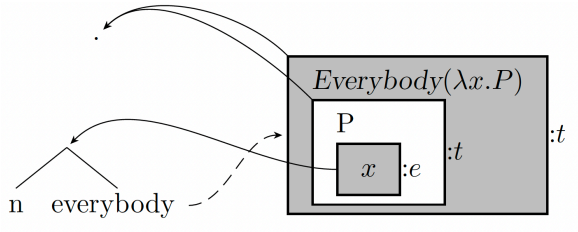


We now have an account for single *wh*-NPs that prepose to the position of their scope.

The other type we will look at here is *in situ*, exemplified by *wh*-NPs in Japanese and many other languages, as well as quantifiers in English. A type we will not consider here are multiple *wh*-questions, which can move one *wh*-phrase, all of them, or none of them; for an analysis of all these types within LFG+glue see Mycock (2006).

The account of *in situ* constructions, both *wh*-questions in Japanese and quantifiers in English, is a simpler version of the analysis above of *wh*-questions, on the basis of our earlier treatment of quantifiers in section 1. A possible SLE for *everybody* is (30), where the upper syntactic component of the SLE has no syntactic connections to the lower one:

(30)



Condition (8d) on acceptable hookups will cause the node of the upper part of the syntactic specification to be identified with some syntactic structure that includes the instance of *everybody* in the lower part of the syntactic structure, so that the x will be properly bound.

This has the behavior of the earliest version of quantification in glue, using universal linear quantification, as in Dalrymple et al. (1997), which does not include any syntactic restrictions on quantifier scope. If this analysis is accepted, the actual restrictions on quantifier scope that have been observed, as discussed for example in Gotham (2021), have to be treated as semantic in origin. This seems possible, but has not been worked out. Gotham provides a modification of the glue that allows them to be described in glue, but uses a deductive formulation, which I have not worked out how to implement in the box notation used here.

The case of *in situ wh*-phrases is similar to *in situ* quantifiers, with a difference that semantics is plausibly involved in determining their scope. For example narrow scope is strongly preferred or required for the quantifier in (31a), but wide scope is required for the *situ* echo question in (31b).

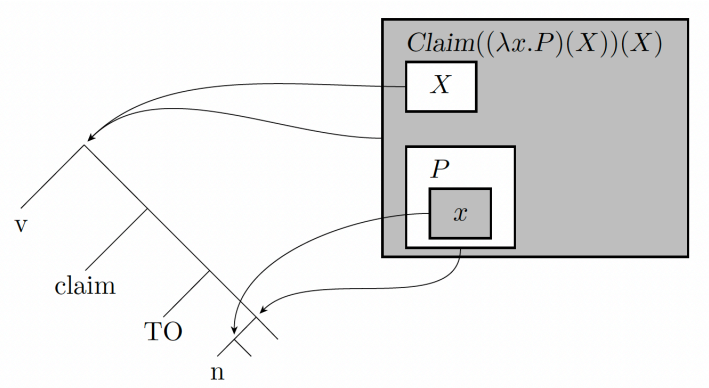
(31) a. Somebody thinks that John gave a treat to every dog

b. Somebody thinks that John gave a treat to WHICH dog??

I propose that this be accommodated by having the output type of the *wh*-constructions be different from that of the quantifiers: the latter would be of type $(e \multimap t) \multimap t$, while the type of the former would be $(e \multimap t) \multimap ct$, '$ct$' being the type of a 'conversational turn', only well-formed as a full independent conversational turn, expecting a response from the interlocutor.

A final observation I will make is that glue can implement a version of the obligatory control as movement of Hornstein (1999). Given the previous proposals, the SLE for *claim* might be:

(32)



The links from the unshaded box labelled P and the shaded box labelled x construe the TO-complement as the predicate $\lambda x.P$, which is the first argument of the meaning of *claim*. We assume that other factors, not discussed here, force the subject of the TO-complement to move to the regular subject position for external arguments, here treated as specifier of *DO*, whereupon the SLE for *DO* will deliver this meaning to the location linked to the unshaded box labelled X, so that it functions both as the agentive argument of *claim* and the (possibly derived) subject argument of the TO-complement. Glue semantics therefore allows us to implement Hornstein's analysis of obligatory control without any additional mechanism of FORM COPY. The semantic analysis is a Hopf-algebraic Minimalist version of the LFG analysis of Asudeh (2005).

4 Conclusion

I have presented a technique for applying LFG's glue semantics to Hopf-Algebraic Minimalist derivations. Unlike the proposal in Marcolli et al. (2024), it does not use traces, but Marcolli et al. (2024) do not give a full account of how traces would work for semantic interpretation. A significant difference between their formulation and this one is that this one uses the 'deletion' coproduct Δ^{ρ} (Marcolli et al. 2024: 29), which has the property of leaving a residue that is not binary branching, since the mother of an extracted node will be left with a single daughter if only one is extracted, as would be the case in a linguistically realistic example. This is not satisfactory if we insist on a Hopf algebra of binary trees, although the nonbranching node can be removed to produce the Δ^{d} coproduct. However the system is also supposed to contain an operation of 'Form Set', which combines k trees (assumed to be binary) into a k-ary branching tree, for arbitrary

k. Form Set is claimed not to be Merge (Marcolli et al. 2024: 141-142), on various grounds, but Merge must apply to trees containing k-ary subtrees, so there is no clear basis for avoiding subtrees where $k = 1$. The Δ^ρ version of the coproduct also has a relatively simple proof of coassociativity, provided by Foissy, and automatically has an antipode, and is therefore immediately a Hopf algebra, rather than requiring some further adjustments (Marcolli et al. 2024: 36).

Another issue, noted by Annie Zaenen, is how this approach is different from TAGs, which have already been combined with glue by Findlay (2019). The answer is that TAGs derive both the semantics and the overt forms from a single structure, subtrees spanning multiple lexical items, while in both the present proposal and LFG+glue, semantics and overt form come from different structures. However the present proposal and Andrews' version of LFG+glue share with Findlay's TAGs the property that semantic interpretation applies to constellations of items that can be regarded as 'lexical items' of some kind, but lacking certain features of overt lexical items, such as linear ordering.

Appendix A: From the box notation to conventional proof-nets

A conventional proof net presents a deduction as a pair of forests of (ordered/planar) syntax trees of formulas, the premises written to the left of a '⊢' (turnstile) symbol, the conclusions to the right, with the leaves connected by axiom links, obeying certain rules. For an intuitionistic deduction, there is only a single formula on the right, and in glue, we normally want an atomic formula in that position. In the version of de Groote (1999), the nodes of these trees are decorated with polarities, and also algebraic expressions which play a role in the algebraic correctness criterion. Perrier (1999) then replaces the abstract algebraic expressions with lambda-calculus formulas, producing a semantic reading procedure as well as another version of the correctness criterion (which guarantees that a 'proof structure' that satisfies it represents a class of valid deductions, and is therefore a proof-net). Beyond these publications, more on proof-nets can be found in Troelstra (1992), Crouch & van Genabith (2000), and Moot 2002, among many other sources.

The correspondence between the present box notation and conventional proof-nets is as follows, taking premises as having positive polarity (negative is also used in the literature, although I think positive is more intuitive for semantics):

1. Each box represents an atomic formula or a 'maximal comb' of implications, positive if shaded, negative if unshaded.

2. The outermost shaded boxes are the premises, the conclusion is implicit; one can think of the shaded box with no overt link as implicitly axiom-linked to a non-overt unshaded box representing the conclusion.

3. The correspondence to conventional proof-nets is simpler if we do not use the upper-case substitution variables, but instead list the sub-boxes in the same left to right order as the arguments they represent appear for a shaded positive box, or the lambda-bound variables for an unshaded negative box.

It is also worth observing that if the basic semantic values of the boxes (assigned as 'outer values' to the premises in the de Groote/Perrier formulation) are just atomic symbols, the 'semantic reading' procedure does nothing other than implement the correctness criterion: the original boxes and links are just as good as a notation as the expressions in (linear) lambda calculus that the semantic reading procedure produces.

Appendix B: A lightning sketch of Hopf algebras

Below is a rapid sketch of Hopf algebras (actually, only of bialgebras, the last step on the way to Hopf algebras), with their application to tree-based syntax, aimed at the level of somebody who might have done basic abstract algebra, but has not used it much. For more extensive discussions at various levels, see Ardila (2012), Foissy (2013), Marcolli et al. (2024) and further references cited there.

A Hopf algebra is a vector space with some additional structure:

(33) a. a 'product' whereby the vectors can be multiplied

b. a 'coproduct' whereby the vectors can be in a sense disassembled

c. a 'unit' which maps scalars onto vectors, allowing them to to function as if they were vectors

d. a 'counit' mapping the vectors onto the scalars

e. an 'antipode', which is a rather abstract sort of inverse, and is automatically present in the Hopf algebra we will be using

All this is subject to various rules.

If a vector space has a product and a unit obeying the rules, it is an 'algebra', in one of several uses of this term. An algebra in this sense that all readers

would have encountered is secondary school polynomials in variables such as x, y, etc. which can be multiplied by numbers (real, rational, or complex) with the results added together (e.g. $2xy + 14z^2$). A natural basis for this vector space is all products of the variables. Unlike the bases of the geometry-based vector spaces normally introduced in undergraduate abstract algebra courses, this basis is denumerably infinite if the set of variables is denumerable (but always infinite since the variables can be multiplied by each other without limit), but the vector space should contain only finite sums of the basis vectors multiplied by the scalars (otherwise certain things become much more difficult).

In the application to Minimalism:

(34) a. The simplest usable scalars are the rational numbers (needed for the characterization of Minimal Search).

b. The natural basis is workspaces, most easily thought of as sets of disjoint, graph-theoretical, unordered (planar) trees, generally referred to in the mathematical literature as forests.

c. The product is disjoint union of workspaces/forests. This is commutative, although that is not required for algebras (but true by stipulation for the secondary school polynomials).

d. The unit maps the scalar r onto $r\mathbf{1}$, where $\mathbf{1}$ is the empty workspace (the unit of the product). This product is often written as $\sqcup$. The secondary school polynomial algebra is different, since the variables have no substantive product, only a formal one.

As with the secondary school polynomial example, the vectors are constituted only by finite sums.

If we add to the algebra a coproduct and a counit, we get a bialgebra; for the bialgebra of trees, the coproduct takes each tree in the workspace, and produces every way of pulling (possibly total, or empty) subtrees out of that tree, paired with the remainder that is left. Unfortunately, the pairing takes the form of a tensor, which cannot be adequately explained here, but Gowers (2024) is a popular resource.

However for a very sketchy overview, suppose we have two vector spaces V and W, with bases E and F respectively. The tensor product $V \otimes W$ is the space of vectors we can construct by using the members $<e, f>$ as a basis (these normally written as $e \otimes f$), and imposing equalities which have the following effects:

(35) a. The function $\Phi : V \times WV \otimes W$ that takes the members of $V \times W$ onto their equivalence classes as determined by the equalities is bilinear (linear in each input considered separately while the others are held constant at arbitrary values).

b. For any bilinear function $fV \times WZ$, there is a unique linear function $\tilde{f} : V \otimes WZ$ such that $f = \tilde{f} \circ \Phi$.

(35b) is the 'universal property' of the tensor product, and can be seen as a way of saying that Φ is the most general bilinear function on $V \times W$.

A useful consequence is that if a binary operation on a vector space $o : V \to V \times V$ is bilinear, it corresponds to a unique linear mapping from $V \otimes V$, typically written as $\tilde{o}$, although people tend to not bother to write the tilde, since it can be supplied as needed from the context. So one might see $\sqcup : W \otimes W \to W$ for the product of the workspace algebra, since the product has to be bilinear if thought of as from $V \times W$.

Now we can look at a very simple case, what the coproduct we use here, written Δ^ρ, does to the tree $[a, b]$, representing trees as square brackettings (which need to be be considered as unordered):

(36) $\Delta^\rho([a, b]) = \mathbf{1} \otimes [a, b] + [a, b] \otimes \mathbf{1} + a \otimes [b] + b \otimes [a]$

Applied to a forest rather than a single tree, the results of all these extractions are added together, and a single extraction can remove any number of items from any number of trees, as long as none of the removed items are contained in any others (a restriction intriguingly reminiscent of the concept of 'analyzability' in classic Transformational Grammar).

A small increase in the complexity of a tree or forest produces a big increase in the complexity of the coproduct, which needs to obey a condition of 'coassociativity', making the entire thing rather confusing; we will leave further exposition to the mathematicians. The counit also has to obey a compatibility condition with the coproduct, and if all the rules are satisfied, the result is a bialgebra. Finally, if there is a further item called an 'antipode', which exists automatically for the bialgebra of unordered trees with Δ^ρ as the coproduct (some of the other coproducts discussed in Marcolli et al. (2024) need some adjustments to get an antipode), the result is a Hopf algebra. However, so far, the antipode plays no role in this formalization of syntax.

Acknowledgments

The connection of this paper to Dick Crouch's work is that he has been a leader in introducing new ideas from logic and mathematics to linguistics, and here I try to apply one of the things he has worked on, glue semantics, to this new recent introduction.

I would like to acknowledge the very helpful comments of Ash Asudeh, Annie Zaenen, and an anonymous reviewer. I have also benefited from from being able to present this material to an online seminar conducted by Andras Kornai in 2023. Errors and obscurities are of course entirely mine.

References

Andrews, Avery D. 2007. Generating the input in OT-LFG. In Jane Grimshaw, Jane Simpson, Tracy Holloway King, Joan Maling, Christopher D. Manning & Annie Zaenen (eds.), *Architectures, rules, and preferences: A festschrift for Joan Bresnan*, 319–340. CSLI Publications.

Andrews, Avery D. 2008. The role of PRED in LFG+glue. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG08 conference*, 46–76. CSLI Publications. https://typo.uni-konstanz.de/lfg-proceedings/LFGprocCSLI/LFG2008/lfg08.html.

Andrews, Avery D. 2010. Propositional glue and the correspondence architecture of LFG. *Linguistics and Philosophy* 33. 141–170.

Andrews, Avery D. 2019. A one-level analysis for Icelandic Quirky Case. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG19 conference*, 24–27. CSLI Publications. https://typo.uni-konstanz.de/lfg-proceedings/LFGprocCSLI/LFG2019/index.shtml.

Ardila, Frederico. 2012. *Hopf algebras*. https://fardila.com/Clase/Hopf/lectures.html.

Asudeh, Ash. 2005. Control and semantic resource sensitivity. *Journal of Linguistics* 41(3). 465–511.

Asudeh, Ash. 2022. Glue semantics. *Annual Review of Linguistics* 8. 321–341.

Asudeh, Ash. 2023. Glue semantics. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*, 651–697. Language Science Press. DOI: 10.5281/zenodo.10185964. https://doi.org/10.5281/zenodo.10185964.

Asudeh, Ash & Richard Crouch. 2002a. Coordination and parallelism in glue semantics: Integrating discourse cohesion and the element constraint. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG02 conference*, 19–39. CSLI Publications. https://typo.uni-konstanz.de/lfg-proceedings/LFGprocCSLI/LFG2002/lfg02.html.

Asudeh, Ash & Richard Crouch. 2002b. Derivational parallelism and ellipsis parallelism. In Line Mikkelsen & Christopher Potts (eds.), *WCCFL 21 proceedings*, 1–14. Cascadilla Press.

Bernódusson, Helgi. 1982. *Ópersunólegar Setningar [impersonal sentences]*. University of Iceland. (MA thesis).

Chomsky, Noam, T. Daniel Seely, Robert C. Berwick, Sandiway Fong, M.A.C. Huybrechts, Hisatsugu Kitihara, Andrew McInnerney & Yushi Sugimoto. 2023. *Merge and the strong Minimalist thesis*. Cambridge University Press.

Crouch, Richard & Josef van Genabith. 2000. *Linear logic for linguists*. http://www.coli.uni-saarland.de/courses/logical-grammar/contents/crouch-genabith.pdf.

Dalrymple, Mary (ed.). 1999. *Semantics and syntax in Lexical Functional Grammar: The resource logic approach*. The MIT Press.

Dalrymple, Mary. 2001. *Lexical Functional Grammar*, vol. 34 (Syntax and Semantics). Academic Press.

Dalrymple, Mary, Vineet Gupta, John Lamping & Vijay Saraswat. 1999. Relating resource-based semantics to categorial semantics. In Mary Dalrymple (ed.), *Semantics and syntax in Lexical Functional Grammar: The resource logic approach*, 261–280. The MIT Press.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell III & Annie Zaenen (eds.). 1995. *Formal issues in Lexical-Functional Grammar*. CSLI Publications.

Dalrymple, Mary, John Lamping, Fernando Pereira & Vijay Saraswat. 1997. Quantification, anaphora and intensionality. *Logic, Language and Information* 6. Appears with some modifications in Dalrymple (1999), pp. 39–90, 219–273.

de Groote, Philippe. 1999. An algebraic correctness criterion for intuitionistic multiplicative proof-nets. *Theoretical Computer Science* 224(1-2). 115–134.

Findlay, Jamie Y. 2019. *Multiword expressions and the lexicon*. Oxford University. (Doctoral dissertation).

Foissy, Loïc. 2013. *An introduction to the Hopf algebras of trees*. https://www2.mathematik.hu-berlin.de/~kreimer/wp-content/uploads/Foissy.pdf.

Gotham, Matthew. 2018. Making logical form type-logical: Glue semantics for Minimalist syntax. *Linguistics and Philosophy* 41(5). 511–556.

Gotham, Matthew. 2021. Approaches to scope islands in LFG+glue. In Miriam Butt, Jamie Y. Findlay & Ida Toivonen (eds.), *Proceedings of the LFG21 conference*, 146–166. CSLI Publications. https://typo.uni-konstanz.de/lfg-proceedings/LFGprocCSLI/LFG2021/lfg21.html.

Gowers, Tim. 2024. *How to lose your fear of tensor products*. Undated webpage accessed in 2024. https://www.dpmms.cam.ac.uk/~wtg10/tensors3.html.

Halvorsen, Per-Kristian & Ronald M. Kaplan. 1988. Projections and semantic description in Lexical-Functional Grammar. In *Proceedings of the international conference on fifth generation computer systems*, 1116–1122. Also published in Dalrymple, Kaplan, Maxwell & Zaenen (1995), pp. 279–292. Institute for New Generation Computer Technology.

Harley, Heidi. 2014. On the identity of roots. *Theoretical Linguistics* 40(3–4). 225–276.

Heim, Irene & Angelika Kratzer. 1998. *Semantics in generative grammar*. Blackwell.

Hornstein, Norbert. 1999. Movement and control. *Linguistic Inquiry* 30. 69–96.

Jackendoff, Ray. 1973. The base rules for prepositional phrases. In Stephen R. Anderson & Paul Kiparksy (eds.), *A festschrift for Morris Halle*, 345–357. Holt, Rinehard & Winston.

Kay, Paul & Charles J. Fillmore. 1999. Grammatical constructions and linguistic generalizations: The what's X doing Y? construction. *Language* 75(1). 1–33.

Kokkonidis, Miltiadis. 2008. First-order glue. *Journal of Logic, Language and Information* 17. 43–68.

Lakoff, George. 1970. Global rules. *Language* 46. 627–639.

Lamarche, Francois. 1994. *Proof nets for intuitionistic linear logic 1: Essential nets*. Technical Report, Imperial College London.

Marantz, Alec. 1984. *On the nature of grammatical relations*. MIT Press.

Marcolli, Matilde. 2023a. *A mathematical model of syntactic merge*. Lecture at University of Utrecht, June 25, 2023. https://www.youtube.com/watch?v=JPdjRaU4JNU&t=2027s.

Marcolli, Matilde. 2023b. *A mathematical model of syntactic merge*. Lecture series at MIT. https://www.youtube.com/watch?v=7Gtpc_E1Tfw.

Marcolli, Matilde. 2024. *Mathematical models of generative linguistics (Ma191c spring 2024)*. CalTech course materials. https://www.its.caltech.edu/~matilde/LinguisticsMa191Spring2024.html.

Marcolli, Matilde, Robert Berwick & Noam Chomsky. 2023. *Old and new Minimalism: A Hopf algebra comparison*. https://arxiv.org/abs/2306.10270.

Marcolli, Matilde, Noam Chomsky & Robert Berwick. 2023a. *Mathematical structure of syntactic merge*. https://arxiv.org/abs/2305.18278.

Marcolli, Matilde, Noam Chomsky & Robert Berwick. 2023b. *Syntax-semantics interface: An algebraic model*. https://lingbuzz.net/lingbuzz/007696.

Marcolli, Matilde, Noam Chomsky & Robert Berwick. 2024. *Mathematical structure of syntactic merge*. Unpublished manuscript, CalTech. https://www.its.caltech.edu/~matilde/MergeMCB-MITPress-LI.pdf.

Moot, Richard. 2002. *Proof-nets for linguistic analysis*. University of Utecht. (Doctoral dissertation).

Mycock, Louise. 2006. *The typology of constituent questions: A Lexical-Functional Grammar analysis of 'WH'-questions*. University of Manchester. (Doctoral dissertation).

Partee, Barbara H. 2009. Do we need two basic types? *Snippets: Special issue for Manfred Krifka* 20. 37–41.

Partee, Barbara H. & Vladimir Borschev. 2004. Genitives, types, and sorts. In Jiyung Kim, Yury A. Lander & Barbara H. Partee (eds.), *Possessives and beyond: Semantics and syntax*, 29–43. UMass GSLA. http://people.umass.edu/partee/Research.htm.

Perrier, Guy. 1999. Labelled proof-nets for the syntax and semantics of natural languages. *Logic Journal of the IGPL* 7. 629–654.

Stabler, Edward P. 2010. Computational perspectives on Minimalism. In Cedric Boekx (ed.), *Oxford handbook of linguistic minimalism*, 616–641. Oxford University Press.

Troelstra, Anne S. 1992. *Lectures on linear logic*. CSLI Publications.

Wood, Jim & Alec Marantz. 2017. The interpretation of external arguments. In Roberta D'Alessandro (ed.), *The verbal domain*, 255–278. Oxford University Press.

Part II

Reasoning and Discourse