Glue Semantics for Hopf-Algebraic Mininimalism

Avery Andrews
v1.5, ANU, Feb 2024

The purpose of this paper is use the 'glue semantics' of Lexical-Functional Grammar (LFG) to provide a form of semantic interpretation for the new approach to Minimalist syntax using Hopf algebras presented in Marcolli et al. (2023b,a) and several lectures, especially the 5-lecture series at MIT,[1] which I will collectively refer to as MC&B2023. Glue semantics is essentially Montague-style Semantics piggybacked onto a very limited 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 possibly even cycles, although these have never been generally accepted as necessary in LFG), and are furthermore too flat to directly induce quantifier scope relations. In the case of MC&B2023, if we want to implement Chomsky's idea that External Merge is responsible for Theta-role assignment, the problem is that the syntactic derivation takes the form of a succession of distinct workspaces, so how do we integrate the results of these into a single semantic interpretation? This approach appears to be very different from the one proposed in Marcolli et al. (2023c), and also the MIT lectures, and it is not impossible that it is completely erroneous, but hopefully, in this latter case, having the errors pointed out will be helpful to other 'Ordinary Working Grammarians' trying to come to grips with this material. A possible point in it favor is that it does not depend on the coproduct leaving 'traces', thereby avoiding the labelling problem associated with these, discussed at various points in MC&B2023.

In terms of glue, I will be making two non-mainstream assumptions. The first will be to adopt the proposal of Andrews (2007, 2008, 2019) to introduce meaning-contributions on the basis of the contents of the f-structure, rather than by 'meaning-constructors' in conventional lexical entries, as explained for example in Dalrymple (2001). This permits, among other things, a relatively clean treatment of many kinds of multi-word expressions.[2] The second is to adopt a definite position on something which seems to me to be left insufficiently clarified in MC&B2003, which is how to represent the difference between identical-looking occurrences of lexical items and other

[1]'The Mathematical Structure of Syntactic Merge', Lecture 1 currently at `https://www.youtube.com/watch?v=7Gtpc_E1Tfw`

[2]For a very thorough discussion and analysis of MWEs based on a different approach (integrating TAGs into LFG), see Findlay (2019).

subtructures, such as *Don criticized Joe before Joe criticized Don*. The proposal is that the 'lexical items' and semantically interpretable 'features' of MC&B2003 get unique indexes when taken from the lexicon and used to form a workspace, and these indexes are preserved in successor workspaces. This allows positions in the successor workspaces to be specified by their relationship to the interpreted elements, and thereby serve to constrain the placement of the 'axiom links' that implement semantic assembly. A consequence of this construal is that we build the interpretation by following the course of the derivation, at least partially realizing Chomsky's idea that External Merge (EM) is what is responsible for Theta-role assignment, and have no need for 'traces' and the consequent issue of how to assign labels to them. By contrast,the MC&B2023 proposals for semantics seem to work off the final structures, although I confess to understanding almost nothing about how they are supposed to work.

A final presentational, but non-substantive, innovation is the use of a notation for linear logic proof-nets that is inspired by the 'box notation' of construction grammar as used in Kay & Fillmore (1999). One motivation for this is that many people find the deductive presentation of glue rather intimidating, and perhaps the CxG-inspired notation will make it a bit easier. Another is that I think this partial resemblance to CxG is inherently interesting. In the final section of the paper, I will discuss some limitations of the box notation, which has not been extended to all of the proposed features of glue.

A final observation I will make is that this system is independent of Gotham (2018), which adds glue semantics to an older version of Minimalism, although there is at least one common feature, in the use of LFG-style indexing of lexical items. Gotham also provides a careful presentation of the mainstream formulation of glue using deduction, and of some topics that are central to glue, such as scope of nested quantifiers.

I will first present glue assembly on its own, without reference to syntax, and then show how it can be connected to Hopf algebraic Minimalism. In a final section, I will discuss the relationship between the present notation for glue and the mainstream ones.

1 Glue By Itself

The 'rudimentary' linear logic used by glue, mostly intuitionistic implicational (possibly with some extensions), divides naturally into two halves, the easy half, based on 'implication elimination' (implication-left in the Gentzen sequent system), and the less easy half, based on 'implication introduction' (implication-right in Gentzen sequents). In the application to semantics, the first applies predicates to arguments, the second creates lambda-abstracts.[3] In the immediately following subsection 1.1 we deal with the first half, in 1.2 the second, in 1.3 some formal issues involving types.

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). But we differ from standard CxG notation in having the boxes be shaded:

(1) $\boxed{\textit{Don}}$

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:

(2) $\boxed{\begin{array}{c} \textit{Criticize}(X, Y) \\ \boxed{X} \quad \boxed{Y} \end{array}}$

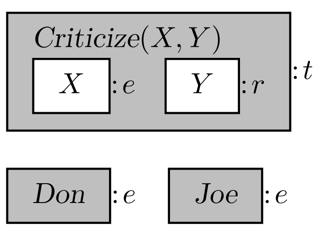
The shading represents 'polarity' in (intuitionistic implicational) proof-nets, as explained by de Groote (1999), with the difference from de Groote that we have taken the shaded boxes to be positive, representing the 'presence of content' to be sent to elsewhere, the unshaded ones to be negative, representing an absence of content, to be remedied by the importation of content from a shaded/positive box. We can also use a variant notation in which there are no upper case variables, but the linear order of the boxes determines the order of application of the predicates to the arguments. There is a further fundamental issue of whether the order of applications, as exists by necessity if curried functions are used, has any linguistic meaning; I would tend to take various 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,

[3] An incomplete but still useful tour of the logical background of glue is provided by Crouch & van Genabith (2000).

as discussed in the literature on Icelandic starting with Bernódusson (1982), and numerous discussions of double object constructions in many frameworks). We can convert from notation with variables to notation without by having the order of the parenthesized variables determine the order of the negative boxes carrying the same variable. A kind of alternative to order would be 'indexed products' as in category theory, but this leads into the problematic area of what the indexes are to be; one could use semantic role lables as in Fillmore or Davidson as the labels, but this lands one in yet another mess of issues as to exactly what labels are to be used. I see this is a fundamentally unsolved problem, which I will not try to solve here.

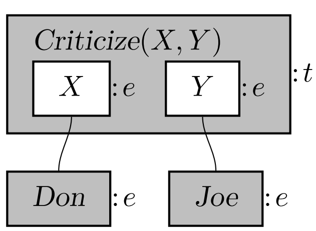
The next elaboration over CxG is that each box has a type, which are the types as usual in basic formal semantics, using simple type theory. Conventionally, people start with types e for 'entity', and t for things at least somewhat like propositions (with truth values), and so shall we. We will write the types of the boxes with colons to their right (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:

(3)



It should now be pretty 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 (3), one of which is depicted below:

(4)



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 would appear that Kay and Fillmore came up with the basic idea of proof-nets for the easier half of intuitionistic implicational logic, but didn't finish the formalization, afaik, 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.[4] We follow de Groote, and, for us, the dynamic graph is composed of two kinds of oriented links:

(5) 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:

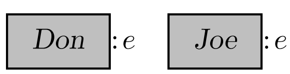
(6) a. Each unshaded box must be axiom-linked to one and only one shaded box.

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

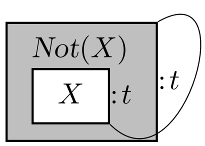
c. The dynamic graph forms a rooted (non-planar) tree.

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

(7) a.



b.



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

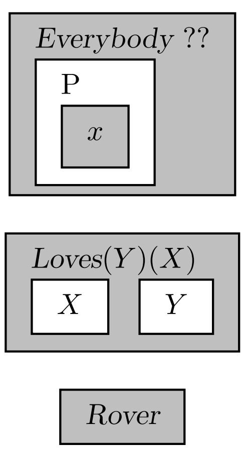
[4] A possible reason for these directionality reversals is perhaps that the more general theory of proof nets is considerably more symmetrical than the intuitionistic one, where there can be any number of premises but only one conclusion.

1.2 Forming Lambda-Abstracts

Montague analysed quantifiers such as *everyone* (incuding what linguists would be likely to call ‘quantified noun phrases’, such as *every sensible person*) as being of the type $(e{\rightarrow}t){\rightarrow}t$, that is, predicates that apply to a complex argument of type $e{\rightarrow}t$ (e.g. *hops*) to produce a truth-value. A problem with this is that such quantifiers also apply 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 our adaptation of the CxG box notation, the essential trick 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 (these are essentially ‘inner values’ in the the algebraic correctness criterion of de Groote (1999), while the semantic labels without their arguments ‘outer values’.)

A preliminary version of the constructors for the meaning of *Rover loves everybody* would be (types omitted because usually are, perhaps to reduce clutter):

(8)

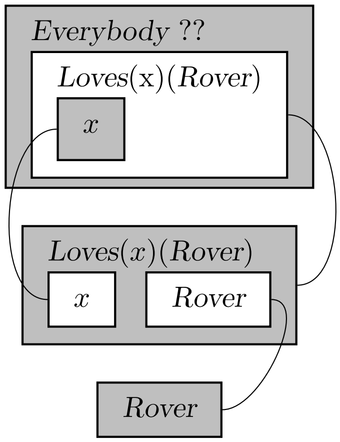


It is clear that to get the intended meaning, we will want to axiom-link the Rover box to the Y box. Then to get a lambda abstract on x, we will want to link the inner shaded x box to the unshaded Y box. But what next?

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

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

(9)



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 substition is:

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

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 as in PROLOG). If the unshaded box contains more than one unshaded 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:

(11) 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 might be 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 correspond to basic types, and in general we will want to be able to have more than one atomic proposition. With implicational logic, we build non-basic propositions with the implication connective, here written '$\rightarrow$', as in Troelstra (1992) (although the 'lollypop' symbol '$\multimap$' is usually used in glue). So the type of a transitive verb would correspond to the proposition $(e{\rightarrow}(e{\rightarrow}t))$, usually written $e{\rightarrow}e{\rightarrow}t$ (or $e \multimap e \multimap t$) by the convention of omitting rightmost parentheses, including outermost.[6]

But in glue, linear logic is not by itself, but functioning as part of an interface, in the literature so far, some kind of typed lambda calculus. Which brings up the question of the relationship between the glue types and the semantic types. So far, I am aware of three possibilities:

(12) 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.f. Partee (2006).

c. The two systems are different.

The first two options are identical in effect, as far as I can determine, so it doesn't 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 $e{\rightarrow}t$, but although this idea seems intriguing, I have not managed to derive any clearly useful results from it. Therefore I will assume either 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 the meaning without resorting to information from the syntax. But syntax is clearly essential to comprehension in many situations, so in the next section, we show how it can be used to constrain semantic assembly.

2 Semantic Assembly and EM

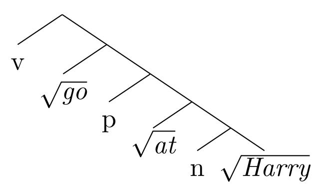
The basic idea here is that of Andrews (2007, 2008, 2019), to use configurations of syntactic items to license the introduction of semantic contributions.

[6]The issue of the presence of linear order in this notation and its apparent absence in ours will be discussed in the third section.

In these papers, this is accomplished by means of a 'Semantic Lexicon', which specifies the intepretations 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 PRED 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. But 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). Hopefully the intended point will survive any necessary modifications to my suggested solutions to these problems.

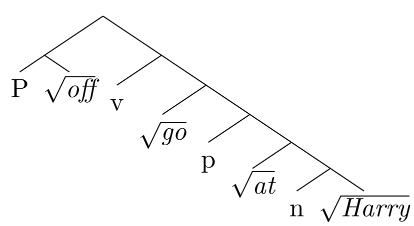
The first of these is what to do about direct complements of verbs and prepositions; I will follow Harley (2014) in putting them as complements to the verb or preposition root under the appropriate categorizer, giving us (12) as the bottom-most portion of the structure:

(13)



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

(14)

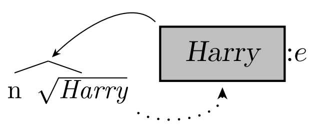


Next, we have the problem of what to do about the 'external argument' *Meghan*. But I am going to defer this, and first indicate how the internal argument *Harry* and various items (here, roots), that are responsible for the meaning of the predicate are handled.

The Andrews papers cited above propose 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 can’t 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.[7] These pointers go from interpetable feature-values to meaning-contributions. The Semantic Lexical Entres (SLEs) also specify what might be called ‘semantic assembly points’ in the syntactic structure, which allow the syntactic structure to constrain semantic assembly; in the present approach, these will go from boxes (both shaded and unshaded) to nonterminal nodes in the syntactic structure.

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

(15)

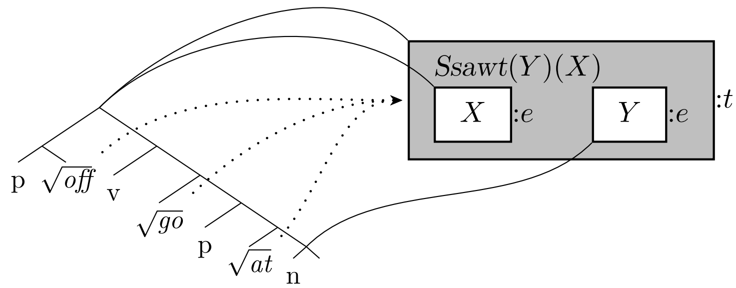


Here we take the category features to be uninterpretable, which allows their presence to be referenced by more than one SLE (quite useful for constraining syntactic combinations), the dotted line from the root $\sqrt{Harry}$ represents the ‘checking off’ link that will cause the instance of this root in the structure to be linked to this instance of the semantic contribution, while the solid links are to the semantic assembly points, which function in a way to be explained 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*, without thinking about how* Y *will feel because of them*, which I will abbreviate as the *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 2nd two to point to the top of the tree, yielding the following SLE:

[7]Dalrymple (2001) discusses cases where it is useful to allow one item to introduce more than one meaning-contribution, although she uses the mainstream version of glue rather than SLEs.

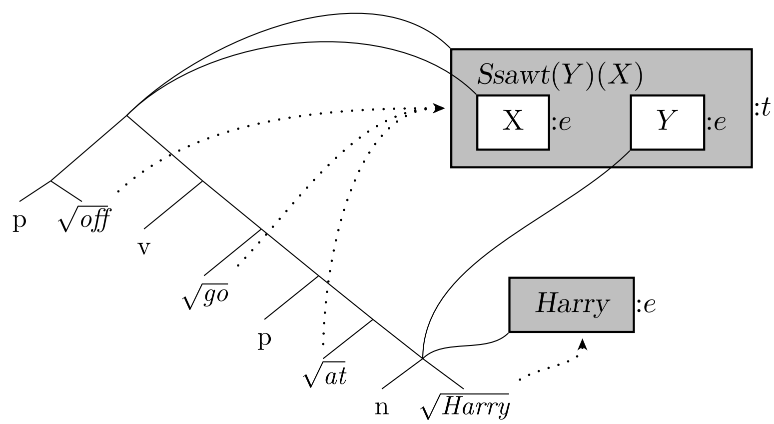
(16)



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, 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.[8]

Next, consider what happens when we apply SLE's to the full tree (14):

(17)



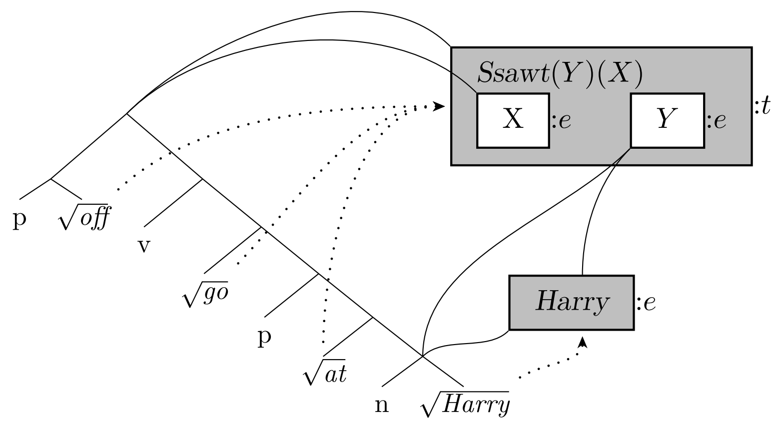
So why

The effect we want to produce is that we can axiom-link the *Harry* box to the Y box but not to the X-box. Visually, we can achieve this by imposing

[8] Observe that in terms of function, an SLE such as the above 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 included in a conventional lexical entry ('codescription'), as discussed by Halvorsen & Kaplan (1988).

the requirement that in order to be axiom-linked, boxes must be associated with the same node in the syntactic structure. Then this hookup is allowed, but the other conceivable one is not:

(18)



This gives us a visual story, but for formalization we want a bit more. In glue semantics, the f-structure locations governing axiom-linking are conceived of as atomic propositions in linear logic, or a component of such propositions, along with the semantic type.[9] This is a straightforward move 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 make some adjustments.

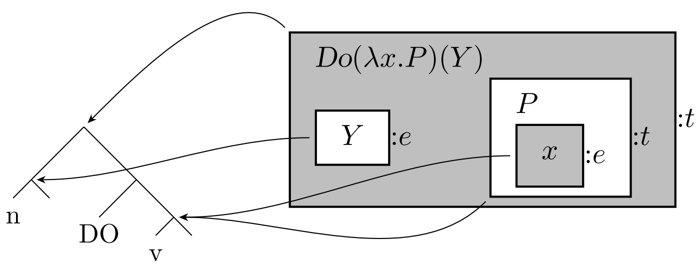
A Hopf-Algebraic Minimalist workspace is produced by a succession of Merge operations applying to an original workspace, which we can think of has having workspace index 0, with a workspace derived by Merge from a workspace with index i having the index $i + 1$. Then I propose that each node in the workspace also gets a unique index, so that in a particular workspace, a node can be identified as an ordered pair of <workspace index, node index>. A glue type-as-proposition can then be a triple with the semantic type included, or, alternatively, we can think of the index pair of being the glue type, with assembly also be required to obey the semantic type restrictions. But, crucially, there is no consistent relationship between the index of a node in one workspace and its intuitive successors in any workspaces derived from that. But under this proposal, the interpretable roots and features do have unique indexes that persist across workspaces, created when they are instantiated from the lexicon (this is a borrowing from LFG).

[9]But any such internal structure within the propositions would be invisible to semantic assembly in glue.

There is however another issue to consider: are the instantiation indexes a property of the terminal nodes that in a sense have lexical content, or a property of the content itself? I will leave this issue unresolved, since as far as I can see now, it has only technical, implementational significance without immediately evident wider consequences. Now we move on to consider how the external argument is going to work.

For this I will use a presumably verbal element which I will call DO, omitting further structure, which takes the external argument in specifier position as one argument, and a predicate in complement position as another. Its SLE looks like this:

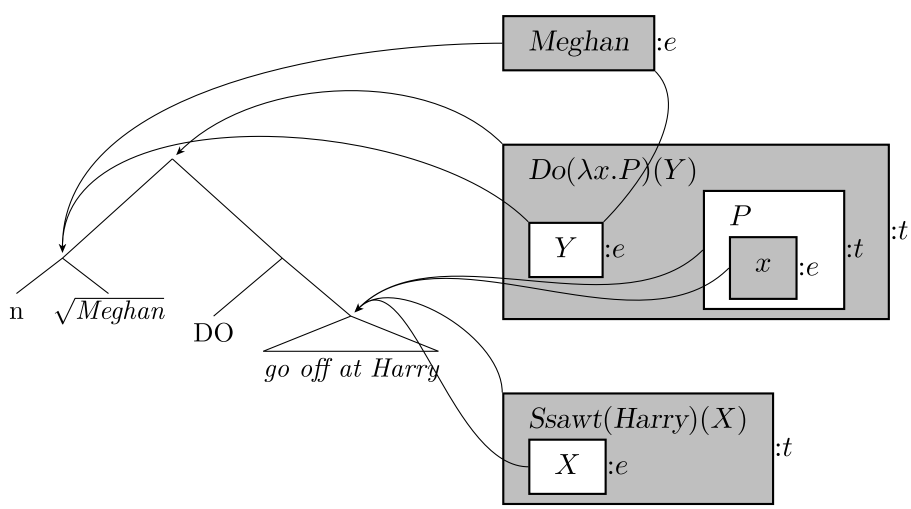
(19)



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 very likely more to it than that.)

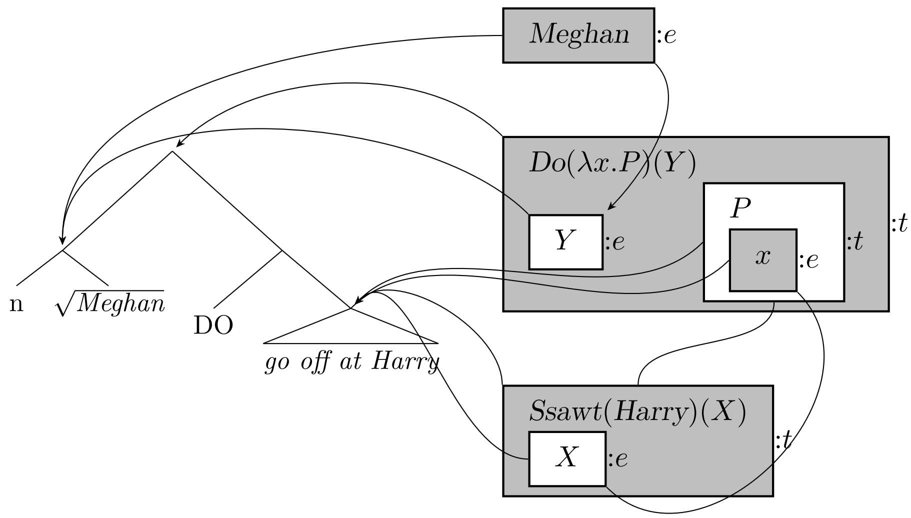
The tree including DO and the external argument will then be (19) below, with the trivial axiom link for *Meghan* added, and the previously assembled structure presented as a block:

(20)



So what about the P box and the predicate material? Here we have four arrows converging on one node. But 2 of them are of one type, and 2 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:

(21)



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 the combine this with *Meghan* to produce the result we want:

(22) $DO(Meghan)(\lambda x.Ssawt(Harry)(x))$.

So what happens if this syntactic structure gets disrupted by IM in a later step in the derivation? The answer is, nothing, because it is the links between the boxes that constitute the semantic assembly. Or, if we were using a overtly deductive formulation, the relevant deductions could all have been done, using the node and workspace indexes present at the time, and their absence from the next would not affect the result derived so far. There will, however, be a problem of dealing with the results of Internal Merge, which we will consider in the next section.

But there is another issue which we will have to deal with when we get to internal merge, so we might as well deal with it now. Our discussion so far has assumed that EM has produced a structure to which two meaning-contributions have been applied, but the first step would have been to combine the categorizer 'n' and $\sqrt{Harry}$ to get something to which we can apply the SLE (15), producing a structure identical to the SLE modulo the details of instantiation. Then with additional applications of EM we can build up something to which (16) can apply, looking almost like (17), but there is a problem in putting them together: in terms of what we said so far, the syntactic attachment link of the *Harry*-SLE was to a node in a different workspace than the one in which we have assembled the materials for the MWE predicate, so without some further work, the requirements for adding an axiom link are not satisifed.

The solution I propose depends on the previous provision that the instantiated interpreted roots and features contain links to any meaning-contributions they license. This permits the syntactic attachment points to be defined, in each workspace, relative to the positions of the interpreted items. So in the case of (15), the attachment point that is the destination of the attachment link from the meaning-contribution will be specified as the mother node of the root. And for (16), the attachment point for the lower argument can be sister of $\sqrt{at}$. Of course the axiom link can't be formed until both of these descriptions designate the same thing, which means they are both applying to the same workspace, so that the <workspace, node> pairs the designate are the same thing, and can be taken as the same atomic proposition in the linear logic (in an explicitly deductive presentation).

A further more general point is this. We have looked at two ways in which the structure (18) can be produced; we would want these two paths to be characterized as essentially the same, which leads us into the territory of 'equivalent proofs', a major theme in proof theory, with an interesting antecedent in early generative grammar, as discussed by Andrews (2021). A google search on "Hopf Algebra" and "Proof Theory" turns up a few things, but it would take some time to absorb them, I think.

But the upper attachment points for MWEs raise the issue of which interpreted feature or root they should be located with respect to, plausible answers being uppermost (in some sense), any, or all. I will not attempt to decide which here.

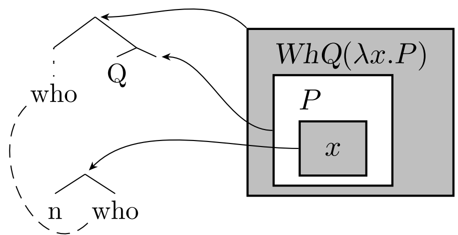
3 Internal Merge

For this section, it is not particularly useful to use an MWE as the main predicate of the example sentence, a simple transitive verb is sufficient, so that's 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 root:

(23)

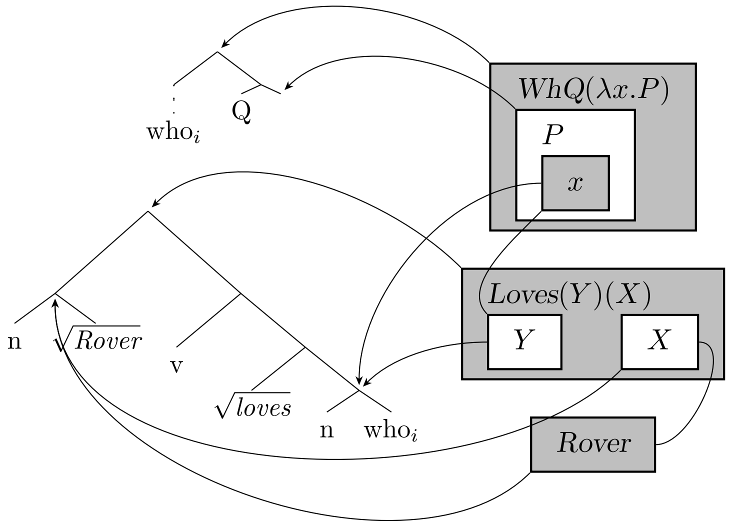


The dashed line connecting the two appearances of *who* is to indicate that they are the same syntactic object, including instantiation index, while the dotted line connecting the upper appearance to the tree-node above it indicates that it can appear inside a containing phrase (pied piping). The nature 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 workspace, but rather, possibly, in successive ones.

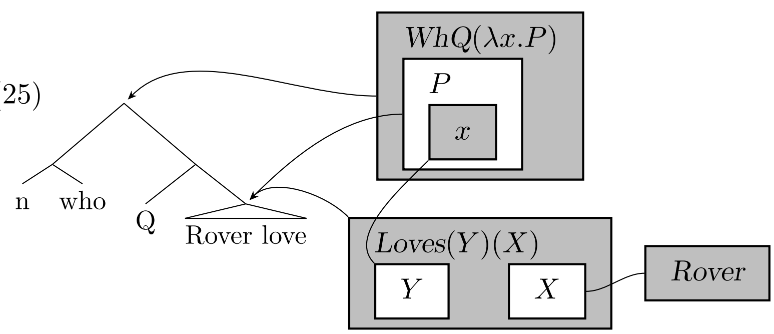
However, for this to work, a meaning-contribution needs to to be able to get a 'partial license' in a workspace where one of its syntactic specifications is satisfied, upgraded to a full licence when all of them are, in a future workspace. *who* can be partially licensed after EM with $\sqrt{love}$ (dotted and dashed lines omitted to reduce clutter, but the appearences of *who* co-subscripted, to indicate that they are have the same instantiation index):

(24)



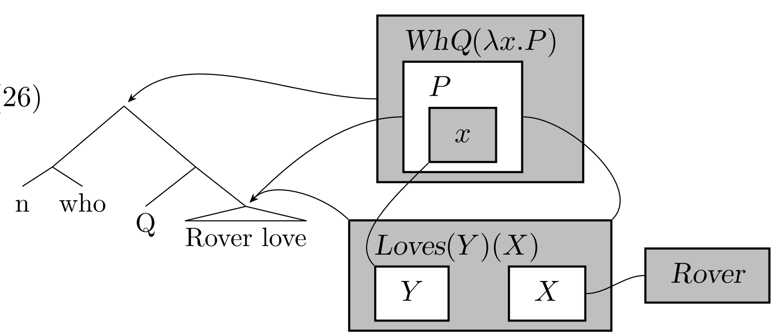
At this point we are stuck, unless we use EM to combine the lower structure (an instantiated tree in a workspace) with Q, and then IM to shift [n who] to initial position, allowing the specifications in the upper tree to be satisfied:

(25)



So now we can add the final link to get the final result:

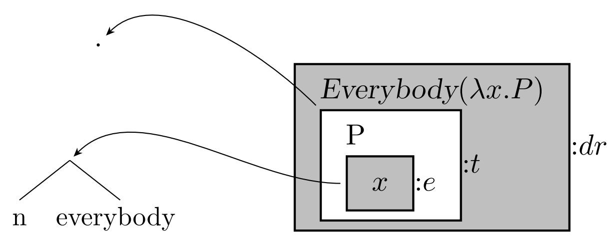
(26)



This treatment accounts for single *wh*-words that prepose to the position of their scope, while *in situ* in languages or constructions without *wh*-preposing (such as echo-questions in English) can be handled in the same way as *many*, to be discussed immediately below. Unaccounted for here are multiple *wh*-questions, either in English, where only one of the *wh*-words preposes, or Greek or Russian, where they all do, and also the situation in Afrikaans (du Plessis, 1977). On the other hand, the workings of glue prevent things from being moved beyond their scope, as discussed by Gotham (2018), a result going back to the early 90s (Dalrymple et al., 1997).

But there is one more thing to deal with, which is wide scope for *in situ Wh*-words, and Quantifiers in English, discussed without reference to syntax in subsection 1.2. Here the approach is essentially the same as that of Andrews (2008), with modifications for Minimalism, similar to Gotham (2018). The approach somewhat resembles what we had for *who*, except that in the simplest version, the upper syntactic component of the SLE shares nothing with the lower one:

(27)



This has the behavior of the earliest version of quantification in glue, using universal linear quantification, as in Dalrymple et al. (1997), but does not include any restrictions on quantifier scope (but this might be correct for certain indefinites, which can take very wide scope, as well as scope limited to certain contexts). Accomodating restrictions on Quantifier Scope raises various problems, discussed from the standpoint of LFG in Gotham (2022).

4 Some further formal issues

In the first subsection, I will explain how the present notation is a somewhat telescoped notation for conventional implicational proofnets. In the second, I will various further features of glue, which may or may not be able to be fit into the notation used here, but should not propose any clear obstacles for a mainstream deductive formulation applied to Minimalism. This section is meant for people who have spent some time with the relevant literature, and so is relatively concise.

4.1 Relating 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 '$\vdash$' (turnstyle) symbol, the conclusions to the right. For an intuitionistic deduction, there is only a single formula on the right, and in glue, we normally want an atomic formula. In the version of de Groote (1999), the nodes of these trees are decorated with polarities, and also algebraic expresions which play a role in his 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) and Moot (2002), among many other places.

The correspondence between the present box notation and conventional proof-nets is as follows, modulo variation in whether the premises are 'positive' or 'negative':

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

2. A row of immediately contained sub-boxes represents the antecents of these implications, in the same order.

3. To make the correspondence simpler, we can provisionally dump the substitution (upper case) variables, and use the linear order. Or the linear order can be derived from the linear order of the substitution variables in the argument positions.

4. By not writing down the expressions with substitution variables, it is made easy to write in the actual values (deGroote's Correctness Criterion produces a proof that there is always a way to do this ... you only have to guess roughly how much space you'll need).

5. Following the linear order of the proof-net trees, the arguments will be ordered from left to right, the lambda-bindings from right to left.

The order-based variant of the notation, with a bit of foresight, provides a reasonable amount of space to write down the semantic values (especially easy using beamer, with the '\onslide' command).

It is also worth observing that if the basic semantic values of the boxes (assigned as 'outer values' to the premises in the deGroote/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. But if the basic semantic values are lambda-expressions, into which the arguments can be substituted without the linearity restriction, then we are specifying a mapping from glue proofs into a cartesian closed category, which is Montague's homomorphism between (some kind of) syntax and (some kind of) semantics. With some versions of the system (without monads or modalities), the glue proof can be seen as a arrow in a Symmetric Monoidal Closed Category, which could also be mapped into a Compact Closed Category to give Compositional Distributional Semantics.

4.2 Extensions of Glue

The discussion and notation so far have used only the most central aspect of glue, implicational intuitionistic linear logic, without universal linear quantification. Glue analyses have proposed and made use of quite a number of extensions, which don't fit the notation introduced here in any worked out way (or, probably in some cases, no way at all), or else interfere with other aspects of the basic system, such as the possibility of mapping into a Compact Closed Category instead of or as well as a Cartesian Closed Category. Here I will list the main ones, with brief comments.

(28) a. Universal Quantification and Modality: The original proposal for quantifiers used linear universal quantification, making it Girard's system F. Gotham (2022) discusses the history of this, finds problems with the 'propositional glue' approach take here, and offers a new solution using modalities.

b. Tensors: These have been widely used the analysis of anaphora, for example by Asudeh (2004, 2012). This approach has been criticized and alternatives advocated by Lev (2007), choosing DRS theory as the best recourse; for more on DRS theory and LFG, see Dalrymple et al. (2018). Tensors do not fit into the present notation very well, but are part of Symmetric Monoidal Closed Categories. It is not clear to me how DRS theory will fare in this connection.

c. Monads: Asudeh & Giorgolo (2016) and Asudeh & Giorgolo (2020) propose to use monads (from Category Theory) to deal with various semantic problems, such as for example the difference between 'at-issue' and 'side-issue' meanings (Potts, 2005). No attempt has been made to integrate these into proof-nets.

By means of the following technique: pretend that all of the monads used in the semantics (as an extension of the lambda-calculus, cf Benton et al. (1998)) are the identity monad, which can then be ignored. The monadic interpretations of semantic formula f returned by glue are then all of the ones that map onto f by converting their monads to the identity monad. These remarks would need a lot of development to turn into a real proposal, but might indicate a way forward.

None of these interfere in any clear way with the project of attaching glue semantics to Hopf Algebraic Minimalism, although they do affect various side issues, such as the status of this notation as more than an introduction to glue, and the applicability of glue to some other approaches to semantics.

References

Andrews, Avery D. 2007. Generating the input in OT-LFG. In J. Grimshaw, J. Maling, C. Manning & A. Zaenen (eds.), *Architectures, rules, and preferences: A festschrift for Joan Bresnan*, 319–340. Stanford CA: CSLI Publications. URL: http://AveryAndrews.net/Papers (Recent Publications (Refereed)).

Andrews, Avery D. 2008. The role of PRED in LFG+glue. In Miriam Butt & Tracy Holloway King (eds.), *The proceedings of LFG08*, 46–76. Stanford CA: CSLI Publications. http://cslipublications.stanford.edu/LFG/13/lfg08.html (accessed 19 Feb 2010).

Andrews, Avery D. 2019. A one-level analysis for Icelandic Quirky Case. In Miriam Butt & Tracy Holloway King (eds.), *The proceedings LFG19*, 24–27. Stanford CA: CSLI Publications.

Andrews, Avery D. 2021. A speculation on what linguistic structures might be. In I Wayan Arka, Ash Asudeh & Tracy Holloway King (eds.), *Modular design of grammar*, 9–21. Oxford University Press.

Asudeh, Ash. 2004. *Resumption as resource management.* Stanford CA: Stanford University dissertation. Retrieved November 15, 2010, from http://http-server.carleton.ca/~asudeh/.

Asudeh, Ash. 2012. *The logic of pronominal resumption*. Oxford University Press.

Asudeh, Ash & Gianluca Giorgolo. 2016. Perspectives. *Semantics & Pragmatics* 9. DOI: http://dx.doi.org/10.3765/sp.9.21.

Asudeh, Ash & Gianluca Giorgolo. 2020. *Enriched meanings: Natural language semantics with category theory*. Oxford University Press.

Benton, P. N., G. M. Bierman & V. C. V. de Paiva. 1998. Computational types from a logical perspective. *Journal of Functional Programming* 8. 177–193. Various earlier drafts available on the web.

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

Crouch, Richard & Josef van Genabith. 2000. Linear logic for linguists. URL: http://www.coli.uni-saarland.de/courses/logical-grammar/contents/crouch-genabith.pdf (checked Feb 1 2024).

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

Dalrymple, Mary. 2001. *Lexical Functional Grammar*. Academic Press.

Dalrymple, Mary, Dag Haug & John Lowe. 2018. Integrating LFGs binding theory with PCDRT. *Journal of Language Modelling* 6.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell & Annie Zaenen (eds.). 1995. *Formal issues in Lexical-Functional Grammar*. Stanford, CA: CSLI Publications. Retrieved Nov 15, 2010, from http://standish.stanford.edu/bin/detail?fileID=457314864.

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

Findlay, Jamie. 2019. *Multiword expressions and the lexicon*. Oxford: Oxford University dissertation.

Gotham, Matthew. 2018. Making logical form type-logical: Glue semantics for minimalist syntax. *Linguistics and Philosophy* 41. 514–556.

Gotham, Matthew. 2022. Approaches to scope islands in lfg+glue. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of lfg2021*, Stanford CA: CSLI Publications. To appear.

de Groote, Philippe. 1999. An algebraic correctness criterion for intuitionistic multiplicative proof-nets. *Theoretical Computer Science* 224. 115–134. Retrieved 15 November, 2010, from `http://www.loria.fr/~degroote/bibliography.html`.

Halvorsen, Per-Kristjan & 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. Tokyo: Institute for New Generation Computer Technology. Also published in Dalrymple et al. (1995), pp 279-292.

Harley, Heidi. 2014. On the identity of roots. *Theoretical Linguistics* 40. 225–276. `http://heidiharley.com/heidiharley/wp-content/uploads/2016/09/Harley2014IdentityOfRootsPublished.pdf`.

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

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

Lev, Iddo. 2007. *Packed computation of exact meaning representations*: Stanford University dissertation. `https://iddolev.com/pulc/material/`. checked Feb 1 2024.

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

Marcolli, Matilde, Robert Berwick & Noam Chomsky. 2023a. Old and new minimalism: a Hopf Algebra comparison. `https://arxiv.org/abs/2306.10270`.

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

Marcolli, Matilde, Noam Chomsky & Robert Berwick. 2023c. Syntax-semantics interface: an algebraic model. `https://lingbuzz.net/lingbuzz/007696`.

Moot, Richard. 2002. *Proof-nets for linguistic analysis*: University of Utecht dissertation. Retrieved November 15, 2010, from http://www.labri.fr/perso/moot/ and http://igitur-archive.library.uu.nl/dissertations/1980438/full.pdf.

Partee, Barbara H. 2006. Do we need two basic types. In Hans-Martin Gaertner, Regine Eckardt, Renate Musan & Barbara Stiebels (eds.), *40-60 puzzles for Manfred Krifka*, Berlin. URL: http://people.umass.edu/partee/docs/. Checked 2 Feb 2024.

Perrier, Guy. 1999. Labelled proof-nets for the syntax and semantics of natural languages. *L.G. of the IGPL* 7. 629–655.

du Plessis, Hans. 1977. *Wh* movementin Afrikaans. *Linguistic Inquiry* 8. 723–726.

Potts, Christopher. 2005. *The logic of conventional implicatures.* Oxford University Press.

Troelstra, A. S. 1992. *Lectures on linear logic.* Stanford CA: CSLI Publications.