Glue Semantics for MCB?

Avery D Andrews
The Australian National University

Nov 2023

Introducing Glue Semantics

1. Developed at Xerox PARC in the 1990s to provide a semantics for LFG that could do quantification properly
2. The basic idea was to use linear logic to enforce the principle that unless there are special provisions, each meaning provided by a lexical item must be used once and once only (you can't understand "I did not eat the last cupcake" as a confession by not interpreting the negative, or interpreting it twice)
3. Early work was based on Girard's System F, but Dalrymple et al. (1999) introduced a major shift to something much more like categorial grammar, with a presentation and application to many examples in Dalrymple (2001).
4. This version can be done with what Valeria de Paiva has called 'rudimentary linear logic', intuitionistic (one conclusion for each premise), using only implication introduction and elimination.

1. Glue semantics works because of the 'Curry Howard Isomorphism', which says that the structure of implicational logic and some of its extensions is the same as that of various forms of the lambda-calculus. In spite of its age, I found Girard et al. (1989) to be quite illuminating for the background.
2. Mainstream glue works by introducing linear logic premises connected to the syntactic structure, called 'meaning constructors', in conventional lexical items (Dalrymple, 2001), or Asudeh (2022) for a recent survey.
3. In my view, glue has been hampered by rather horrible notations, which I have made various attempts to fix; here I will use the latest, inspired by CxG.
4. Mainstream glue also does not provide a very satisfactory account of multi-word expressions, so I will also be using my approach to them from Andrews (2008, 2019). Another way to handle MWEs is Findlay (2019).

1. The present notation is a variant of implicational proof nets, based on work by de Groote (1999) and Perrier (1999). It has a significant limitation in that it doesn't accomodate linear logic tensors very smoothly (these are significantly different from linear algebra tensors!), but I'm not really convinced they are the right tool for what they are used for (anaphora). So this might just be a starter notation to help people get going.
2. But it might also be useful to demonstrate the more often seen deductive notation (or rather one of them, tree-style natural deduction), using what are called 'labelled deduction'. Here the proposition letters represent basic types, and we have implicational/function types represented as $a{\rightarrow}b$ (as opposed to normal formal semantics $<a, b>$, and we deduce with formulas associated with a meaning, coming first and followed by a colon:

Deduction

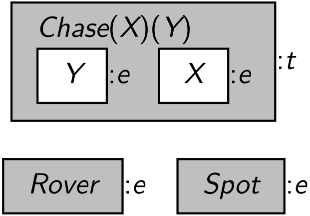
$$\dfrac{Rover : e \qquad \dfrac{Spot : e \qquad Chase : e{\rightarrow}e{\rightarrow}t}{Chase(Spot) : e{\rightarrow}t}\,ImpElim}{Chase(Spot)(Rover)}\,ImpElim$$

In this 'labelled deduction', we have three premises, which combine to produce a single conclusion of type *t*, and we can see that eliminating an implication goes paralell to applying a function to an argument.

It is also evident that if we have to combine all of the premises, using each once only, using the available rule, that there are two possible results, depending on which argument comes first. There's also the issue of which argument gets which semantic role; following Marantz (1984), it is reasonable, but not necessary to apply the least active argument first, so that the meaning delivered by the deduction woule be 'Spot chased Rover'.

The Notation

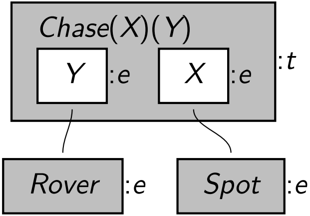
The notation is a rearrangement of proof-nets, inspired by CxG, with the semantic reading as devised by Perrier (1999), based heavily on de Groote (1999). Proof nets are a technique for representing Gentzen-sequent proofs in what is called ‘eta-expanded normal form’, but I’m going to skip the background. Here are the proposed representations for the three premises from the previous slide:



The shaded boxes can be construed as ‘providers’ of meaning, the unshaded ones as consumers aka argument positions, and we connect them with arcs, obeying rules to come.

The Notation

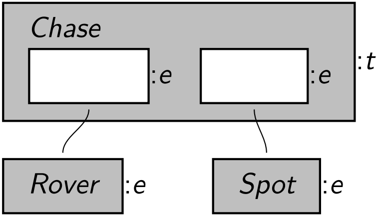
The notation is a rearrangement of proof-nets, inspired by CxG, with the semantic reading as devised by Perrier (1999), based heavily on de Groote (1999). Proof nets are a technique for representing Gentzen-sequent proofs in what is called 'eta-expanded normal form', but I'm going to skip the background. Here are the proposed representations for the three premises from the previous slide:



The shaded boxes can be construed as 'providers' of meaning, the unshaded ones as consumers aka argument positions, and we connect them with arcs, obeying rules to come.

Writing in the Values

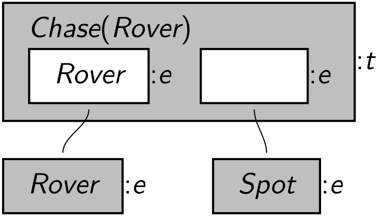
One of the problems with the usual proof-net notations is that there is no convenient place to write in the values, and that is sort of true here as long as we use the substitution variables, but if we depend on linear order, we can write them in as we go:



This is a re-presenation of the semantic reading procedure for proof-nets of Perrier (1999). There will be another value-propagation rule, later.

Writing in the Values

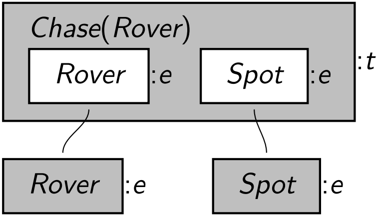
One of the problems with the usual proof-net notations is that there is no convenient place to write in the values, and that is sort of true here as long as we use the substitution variables, but if we depend on linear order, we can write them in as we go:



This is a re-presenation of the semantic reading procedure for proof-nets of Perrier (1999). There will be another value-propagation rule, later.

Writing in the Values

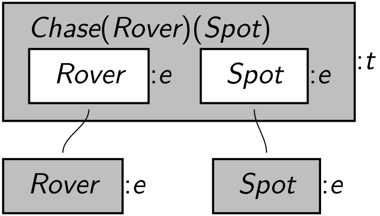
One of the problems with the usual proof-net notations is that there is no convenient place to write in the values, and that is sort of true here as long as we use the substitution variables, but if we depend on linear order, we can write them in as we go:



This is a re-presenation of the semantic reading procedure for proof-nets of Perrier (1999). There will be another value-propagation rule, later.

Writing in the Values

One of the problems with the usual proof-net notations is that there is no convenient place to write in the values, and that is sort of true here as long as we use the substitution variables, but if we depend on linear order, we can write them in as we go:



This is a re-presenation of the semantic reading procedure for proof-nets of Perrier (1999). There will be another value-propagation rule, later.

Hookup Rules 1

The box shading represents ‘polarity’ from de Groote (1999), shaded his ‘negative’, unshaded his positive, althoug he seems to transferring Ben Franklin’s error about electricity to this new domain. For linguistics, it makes more sense to think of shaded as positive ‘producers of meaning’, unshaded as negative, consumers.

There are then rules that ensure that collection of boxes correctly connected by axiom links constitutes a valid linear logic proof of the shaded box that is not directly connected to any unshaded one (de Groote (1999) for proof). So far we have only done Implication Elimination. The two simplest rules are:

1. Every unshaded box must be connected to one and only one shaded box of the same type.
2. Every shaded box except for one must be connected to one and only one unshaded box of the same type

Hookup Rules 2

The third condition requires the notion of ‘dynamic path’, introduced by Lamarche, but with the directionality reversed by deGroote. The dynamic graph consists of directed arcs going along axiom links from shaded to unshaded boxes, and from unshaded boxes to their immediately containing shaded ones. The third condition is:

3. The dynamic graph must form a tree (connected, no cycles)

There will be an additional condition later, when the system includes Implication Introeduction, a.k.a. lambda abstraction.

Some Notes

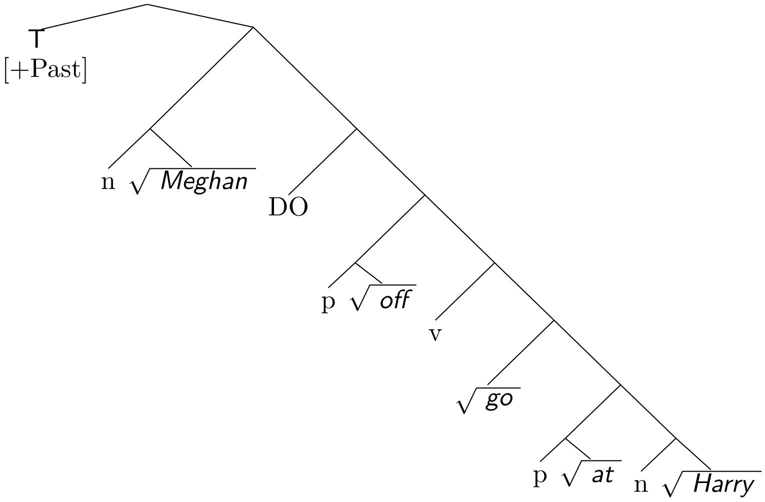
1. Fillmore and Kay would appear to have invented, without knowing it, a notation for the easiest half of 'rudimentary linear logic' with one type.
2. The system is 'constructional' in the sense that a partial hookup/assembly functions just like a single basic unit for further assembly. This property is retained when we complete it.
3. There might be a topological interpretation, in which hookup is acheived by moving the shaded boxes into the unshaded ones they are connected to.
4. The meaning expressions are supposed to be in the ordinary typed lambda-calculus (no linearity restrictions).
5. For assembly, the units function as elements of a closed symmetric monoidal category, so interpretation can be construed as a functor into a cartesian closed category.
6. There is an issue concerning the relations between the types for glue vs those for the meaning language: they are usuall assumed to be the same, but in principle they could be different. Types themselve are worthy of more thought, e.g. Casadio (1988), Partee & Borschev (2004).

Semantic Workspaces?

It should be evident that there are two and only two ways in which our sample collection of constructors can be hooked up, but, in most languages, syntax will exclude one of them (it can be a bit complicated). A possibly useful consequence is that if we can retrieve a collection of meanings from the components of an utterance, there will be a limited number of ways of putting them together, which might help with actual comprehension. This gives us a role for a kind of 'workspace' concept, which might, in fact also be useful for syntax as conceived in Minimalism.

So it is time to push on to how syntax can constrain assembly.

A Sample Structure



General Approach

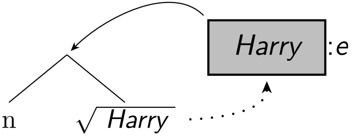
We'll break the interpretation in to three parts, that of the NPs, lower MWE, the upper DO component, which we will take as the account of what 'external arguments' are (in spite of all the haze about this notion, I think there is something to it, and the positing of some kind of upper verb-like element is not the worst possible theory of what that something is).

Intepretation is accomplished by 'semantic lexicon entries' (SLEs), introduced by me for LFG (Andrews, 2007, 2008, 2019), which apply to the structure, introducing a meaning-contribution, and causing certain features to be checked off, the 'interpretable' ones.

I think this approach handles idioms (multi-word expressions) and some other phenomena more cleanly than classic LFG, and with less change to the framework that the TAG-based LFG of Findlay (2019).

Harry

Proper names are easy:



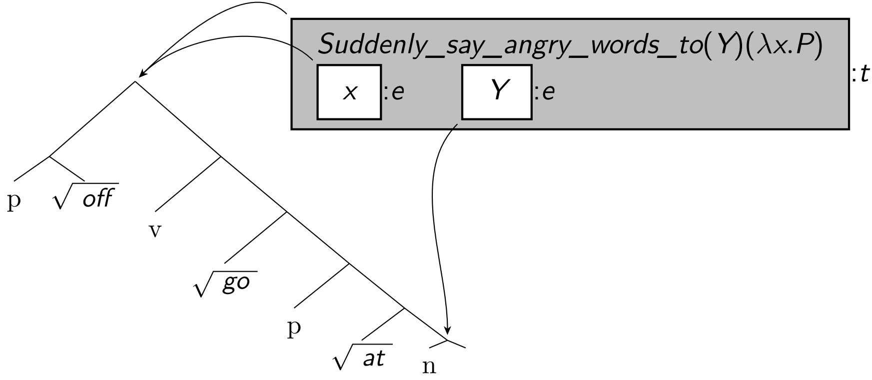
The dotted arrow represents the checking off of the interpretable root, the solid one is a pointer from the meaning contribution to a location in the syntactic structure, which is to guide interpretation.

For things to work, the meaning-contributions need to be introduced as (perhaps contextually modifiable?) copies of the information in the semantic lexicon, otherwise the pointers back to the syntax wouldn't work if the same SLE was used twice.

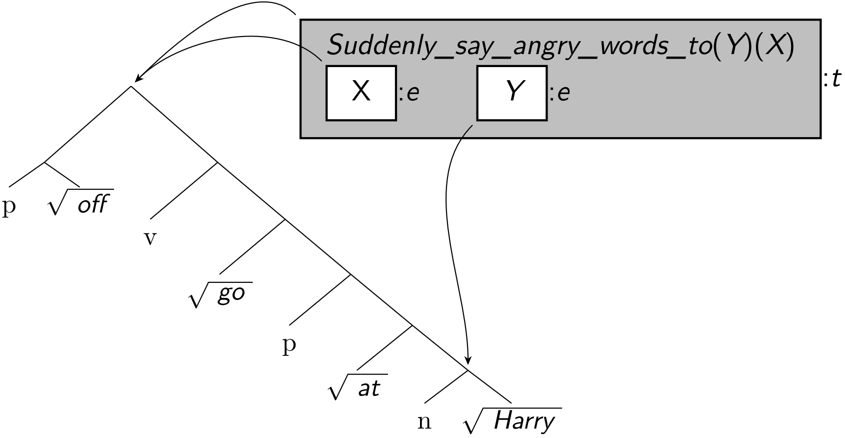
Note that having the category features uninterpreted allows them to be mentioned by multiple SLEs.

The MWE Predicate

The predicate is more complex, involving both an MWE and a second-order argument; I'll explain the MWE and its composition with the object first. Checkoff lines from the three roots omitted for less clutter.

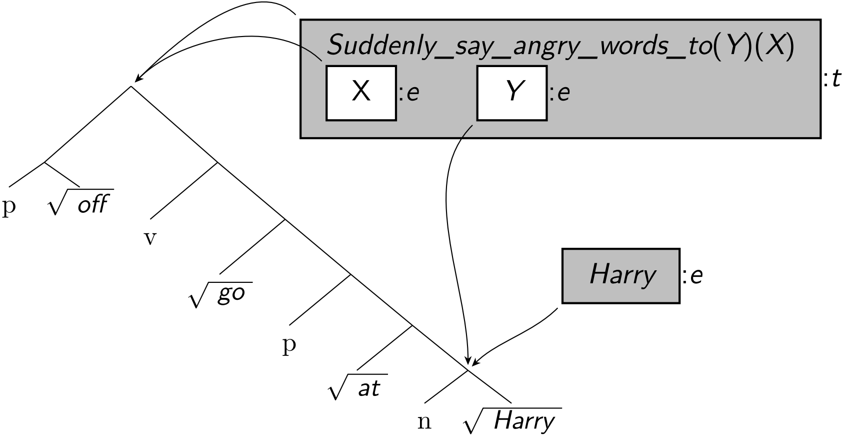


Putting 2 Things Together

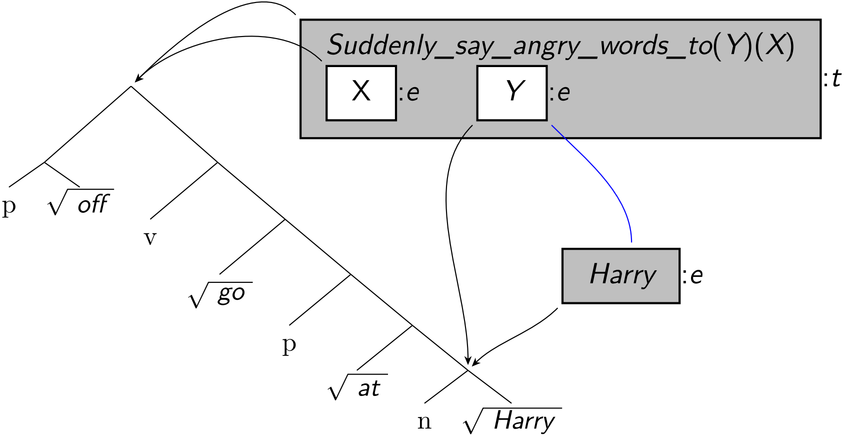


.

Putting 2 Things Together



Putting 2 Things Together



Axiom link in blue.

Yet Another Rule

So, unsurprisingly, the next hookup rule is:

4. A shaded and an unshaded box can only be hooked up if they are connected to the same location in syntactic structure

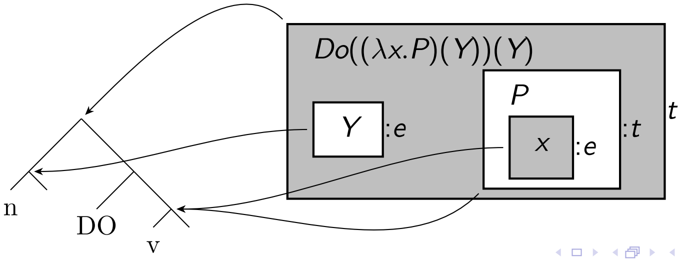
This can perhaps be seen as a kind of extension of the type system.

Now on to the external argument, which will lead to the fourth and last hookup rule

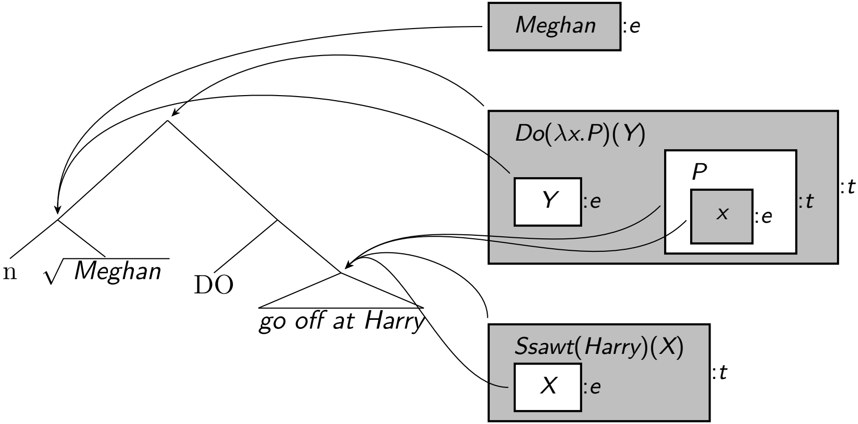
DO

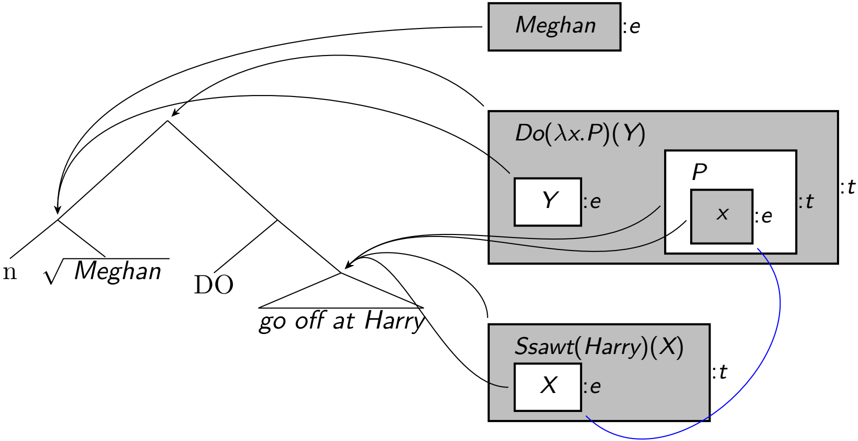
Here we adhere to the proof-net (and glue) tradition, of having links only between basic types (for some motivation which I don't really understand, see Jay & Ghani (1994), but it amounts to using the axiom $p \vdash p$ with only atomic propositions).

Here is the SLE for DO. The lower case variable in the inner solid box is a real, lambda calculus variable, not a dispensible PROLOG-like naive substitution variable.

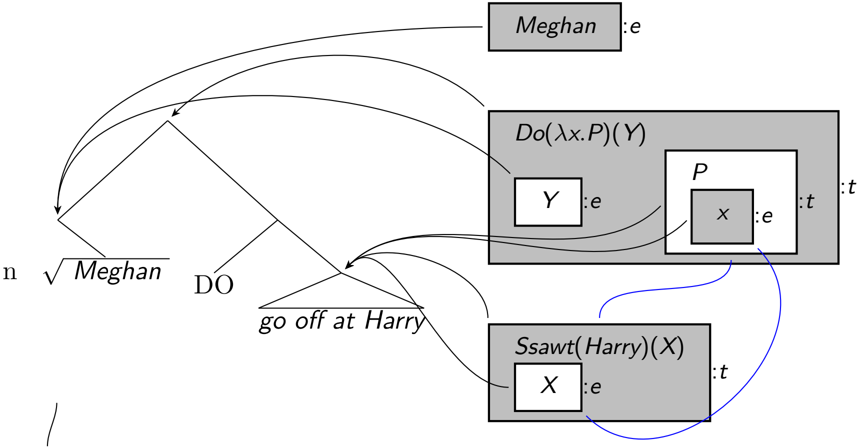


Here we have the lower substructure preassembled.

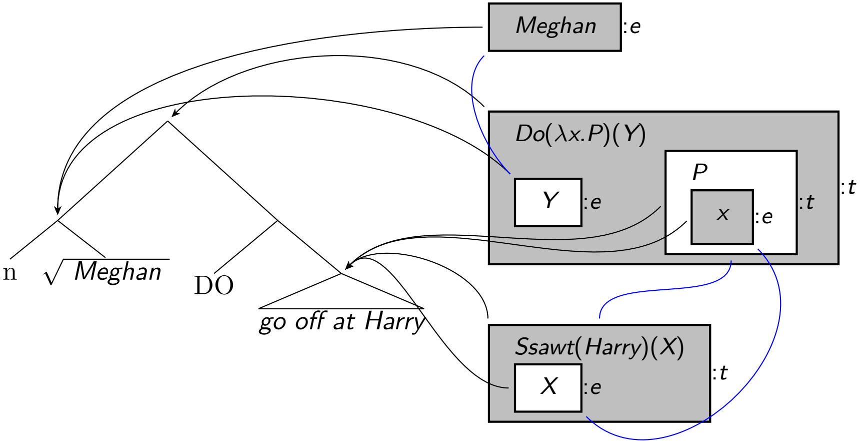




The inner shaded x box gets linked to the unshaded X-box



The lower shaded box gets connected to the *P*-unshaded box in the DO-contribution



After the unexciting final axiom-hookup, we get the final result:
Ssawt(*Harry*)(*Meghan*)

The Last Hookup Rule

We can now formulate the last rule, but we need an additional concept, the 'dynamic path' of Lamarche (1994), with directionality reversed as in de Groote (1999). The reversed dynamic path consists of two kinds of directed links:

1. Axiom links directed from shaded to unshaded boxes
2. Undepicted implicit links from an unshaded box to the immediately containing shaded boz

And now the rule is:

5. The dynamic path from an internal shaded box must pass through the unshaded box that immediately contains it.

The result is that the variables introduced in the inner shaded boxes wind up bound in the values of (all of) their containing shaded boxes. 'Insideness' might help with a topological interpretation.

Its Semantic Reading

And the semantic reading here is that the semantic value of an unshaded box that contains a shaded one is not applied directly as an argument to the predicate of the containing shaded box, but as a lambda-abstract, binding the variable that is the value of the inner shaded box.

So we can think of the substitution variable P in the unshaded box in the slide before last as evaluating to $Ssawt(Harry)(x)$, but the first argument of Do is $\lambda x.Ssawt(Harry)(x)$

The second argument is then the subject, which can be fed to the first argument by the internal workings of the meaning of the DO-predicate.

Interaction with Movement

So what happens if we apply IM to get something like:

(1) At who(m) Meghan go off?

This is a bit odd, but not terrible, and I think its problem is register clash, not syntax as such, since *go off at* is colloquial, ‘pied piping’ of prepositions rather formal. Note for example the contrast with:

(2) On who(m) do we depend?

Where *depend on* is fine in formal registers.

So the question is whether the disruption by the coproduct causes any problem for the semantics. A worry that it might could be triggered by the links from the boxes to the nodes in the syntactic structure, since it is not clear what would maintain the identity of the targets under extraction of a subtree by Δ,

Semantic Structures not in the Workspace

But this is not a problem for the interpretation of MWEs, as well as the assignments of arguments to predicates, because these links do not represent the meaning of a constituent, but only a constraint on how shaded and unshaded boxes can be linked. Marcolli et al. (2023) discuss how the semantic structure is to be rather autonomous from the syntactic one, so the common destination requirement needn't have any further implications for the axiom links once these are establishsed.

A different situation applies with *wh*-operators in English, which are interpreted within the scope they move to, and quantifiers in some other languages, such as Hungarian, which seem to do roughly the same thing. Not knowing Hungarian, I will consider simple constituent questions in English:

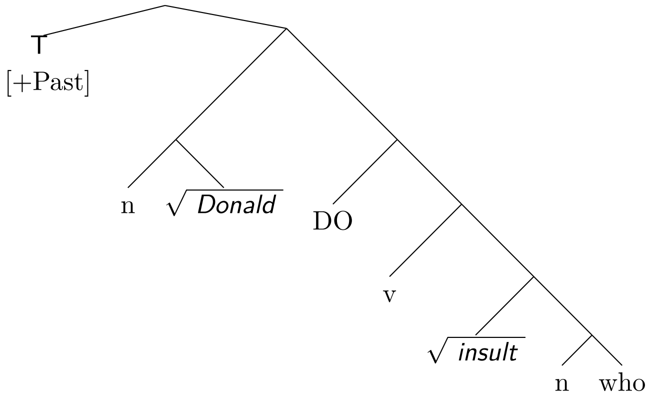
(3) a. Who did Donald threaten?

b. Who did Wapo report that Donald threaten?

For the semantics, I'll assume we want a representation along the lines of

$$Qx(Past(Threaten(x)(Donald))$$

We can assume that successive applications of EM have produced something like:

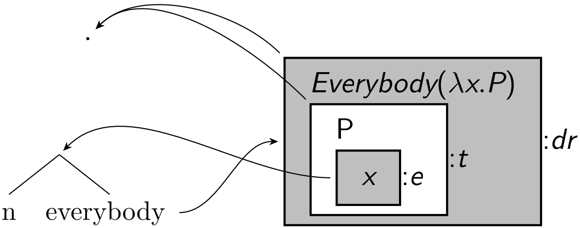


Quantifiers

Before looking at the *wh-* words, it will be helpful to examine the somwhat easier case of generalized quantifiers, in particular the possibility of a GQ in object position taking scope over the entire question:

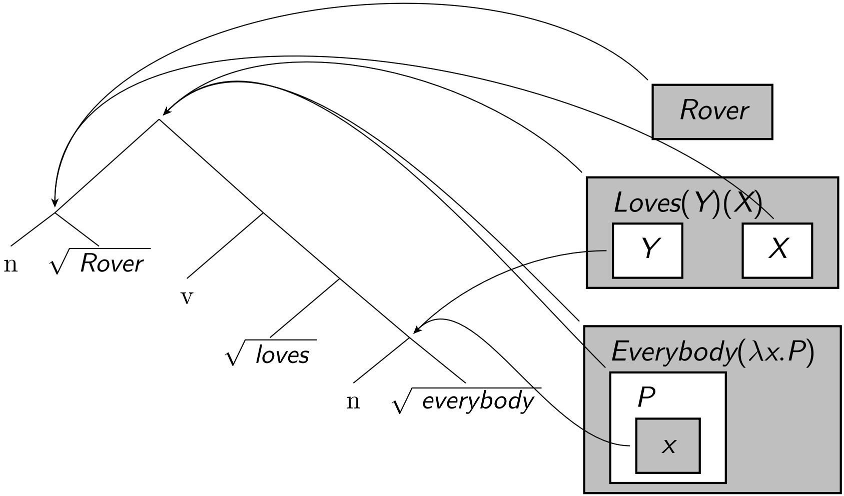
(4) Rover likes everybody

The following SLE will suffice for *everybody*:

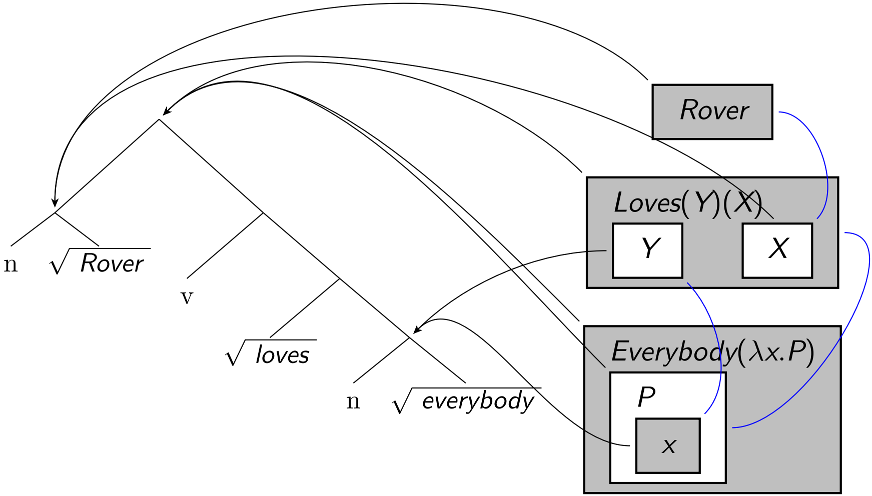


I illustrate with a very (too) simple example:

Supersimple Example



Supersimple Example



More complex cases are discussed in Dalrymple et al. (1997), including

(5) Every representative of a company demoed a product

Which supposedly has 5 readings that glue gets correctly but various other approaches don't (it was a long time ago that I read this, and I've never been very good at complex examples of quantifier scope).

An analysis along these lines will also work for *situ* question words, but ones involving movement, especially obligatory movement, as in English, are more challenging.

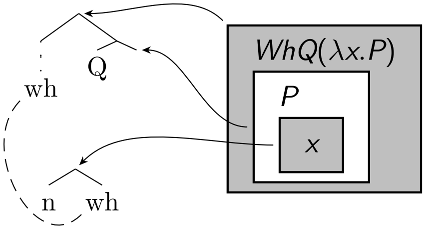
But before taking on the harder case of *Wh*-Question Movement, we need to consider the tricky issue of how this interacts with the connections between the syntactic and the semantic structure.

A Problem with Movement

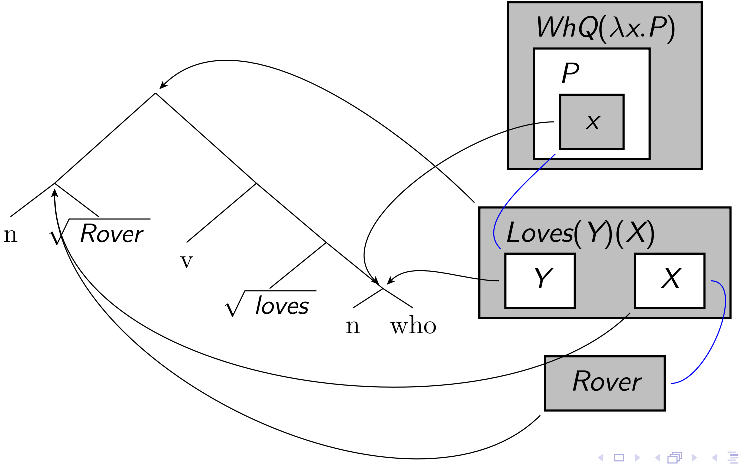
Our account of the syntax-semantics interface makes use of links from interpreted items to meaning contributions, as well as from meaning-contributions to nonterminals, in order to regulate assembly. But how are the targets of these latter links defined? The problem is that in the Hopf-algebraic framework, the identity of individual nodes is not supposed to matter since the outputs of Δ are supposed to be formal sums of isomorphism classes of forests. And set-theoretical extensionality won't help either, since a remainder tree is a different set-theoretical object than the original one.

Therefore, if we use Δ in the implementation of Internal Merge for something like *who does Rover like*, the '*Rover likes*' substructure that appears in the overt from of the question cannot be said to contain 'the same (nonterminal) nodes' as its 'source' '*Rover likes who*'.

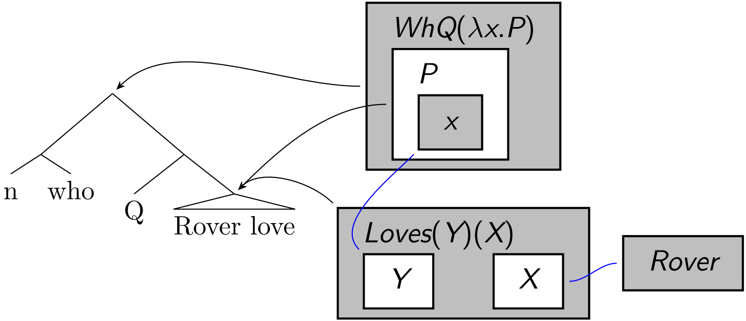
I think there are a number of possible solutions to this; what I will suppose here is that the destinations of the semantic-syntactic links in the SLEs are defined by their relationship to interpreted features, which have their own syntax-to-semantics links. I will also suppose that if the syntactic aspect of an SLE consists of more than one disconnected piece, then they can appear in different workspaces. Therefore I suggest the following, where the dotted line in the tree designates some downward path through it, and the dashed line means that the two connected tokens of '*wh*' designate the same interpreted feature:



So the lower part of our current example could invoke meaning constributions which would assemble partially like this:



Now, if the output of the *Loves* contribution is defined by its relationship to the root that triggers it (great grandmother), same for Q and the *wh*-contribution, after Δ and Merge pull out the *wh*-phrase, these relationships will still exist:



But we need to be able to assume that the '=wh' specifiction for the SLE of *who* is satisfied.

Wrapup and Notes

1. I think the critical assumption here is that one meaning-contribution can have multiple syntactic specifications that can be satisfied in distinct workspaces, but a single axiom-link can only be placed on the basis what is found in a single tree in a single worskpace.
2. Perhaps an alternative could be found on the basis of 'reconstruction' based on the traces that might left by Δ, which can furthermore share mention of a single interpretable feature. But MCB waffle about the traces, so who knows what they really have in mind (latest lectures might help?).
3. In a more formal presentation, it would almost certainly be better to explicitly observe Ron Kaplan's distinction between 'structures', and 'descriptions of structures', the syntactic specifications of SLEs being the latter.

Further Notes

1. Glue is usually formulated a using linear universal quantification for quantifier scope, which makes it Girard's 'System F'.
2. But instead, we can think of that as part of 'instantiation' (attaching the syntactic specification of an SLE or mainstream glue meaning-constructor) to the actual structure), and then the system becomes 'propositional glue' (Andrews, 2010), which is a free Symmetric Monoidal Closed Category. This might be useful for further developments.
3. The box notation might be able to be extended to include monads (Asudeh & Giorgolo, 2020), but probably not tensors (Asudeh, 2012). So if the latter are necessary, the box notation can only be a crutch for beginners. (I'm hoping that monads can displace tensors, but have not yet managed to make it work.)

Relation of Box Notation to Standard Proof Nets

Following the outline of deGroote's algebraic formulation of the correctness criterion, Perrier (1999) showed how to build semantic values for the tree nodes by means of the following 5 rules:

1. The value of the (positive) antecedent of a negative implication is a unique variable (to be lambda-bound later).
2. Values propagate unchanged along axiom links, in the direction of the dynamic graph.
3. The value of the consequent of a positive implication is the value of the implication (a function) applied to the value of its antecedent
4. The value of a negative implication is the value of its consequent lambda-bound by the variable that is the value of the antecedent of the implication (the Correctness Criterion requires the binding to be non-vacuous). The final result gets assigned to the negative node on the rhs of the turnstyle.

Correspondence to Box Notation

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 dump the substitution (upper case) variables, and use the linear order (thereby forgetting for a while about the possibility of a topological interpretation)
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.

Some Readings

1. The anthology for 'Old Glue' Dalrymple (1999), including the especially important Dalrymple et al. (1997) and Dalrymple et al. (1999)
2. The basic linear logic textbook Troelstra (1992), the incomplete but useful Crouch & van Genabith (2000) (covering many approaches to linear logic deduction), and Moot (2002), an extensive discussion of the use of proof nets in categorial grammar.
3. Dalrymple (2001, ch 9), glue analyses of many important constructions, Asudeh (2004, 2012), a glue treatment of anaphora using tensors. Asudeh et al. (2014), glue with Davidsonian analyses.
4. Asudeh & Giorgolo (2020) putting monads into glue.
5. Gotham (2018) Glue semantics for (old) Minimalism. `https://matthewgotham.github.io/research/` has various other formal papers on gluey topics.

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. 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.), *The proceedings LFG19*, 24–27. Stanford CA: CSLI Publications.

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. 2022. Glue semantics To appear in *Annual Review of Linguistics* `http://www.sas.rochester.edu/lin/sites/asudeh/pdf/Asudeh-AR-corrected.pdf`.

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

Asudeh, Ash, Gianluca Giorgolo & Ida Toivonen. 2014. Meaning and valency. In *Proceedings of LFG14*, 68–88. CSLI Publications.

Casadio, Claudia. 1988. Semantic categories and the development of categorial grammar. In Emmon Bach R.T. Oehrle & Deirdre Wheeler (eds.), *Categorial grammar and natural language semantics*, 95–123. Reidel.

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 April 22 2013).

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, Vineet Gupta, John Lamping & Vijay Saraswat. 1999. Relating resource-based semantics to categorial semantics. In Mary Dalrymple (ed.), *Syntax and semantics in Lexical Functional Grammar: The resource-logic approach*, 261–280. Earlier version published in *Proceedings of the Fifth Meeting on the Mathematics of Language*, Saarbucken (1995).

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.

Girard, Jean-Yves, Yves Lafont & Paul Taylor. 1989. *Proofs and types*. Cambridge: Cambridge University Press. Retrieved 15 November, 2010, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.85.5358&rep=rep1&type=pdf.

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

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.

Jay, C. Barry & Neil Ghani. 1994. The virtues of η-expanson. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.613.

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*. Cambridge MA: MIT Press.

Marcolli, Matilde, Noam Chomsky & Robert Berwick. 2023. 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. & Vladimir Borschev. 2004. Genitives, types, and sorts. In Ji yung Kim, Yury A. Lander & Barbara H. Partee (eds.), *Possessives and beyond: Semantics and syntax*, 29–43. Amherst MA: UMass GSLA. URL: http://people.umass.edu/partee/Research.htm.

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

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