Plodding thru (the beginning of) MCB

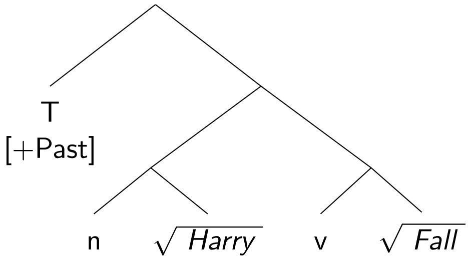
Avery D Andrews
The Australian National University

Nov 2023

Trees & Sets

1. Chomsky has been going on about 'sets' for quite some time
2. I don't get the impression that physicists who use trees for Hopf algebras care about them very much.
3. MCB starts off with quite a lot of confusing waffle about sets vs trees presented as graphs.
4. BUT, in 'non-well-founded set theory', cf the Aczel book from 1988 (which I have poked around in the earlier pages of, but never felt that I attained a proper grasp), we find a procedure for treating directed graphs as 'pictures' of a set. (Acyclic if the set is to be well-founded). `https://plato.stanford.edu/entries/nonwellfounded-set-theory/`.
5. Given that MCB can be formalized with non-cyclic DG's, with no labels on the nonterminals, the DG's can be interpreted as pictures of a set, easily reconstructed from the graph (mother to daughter arrow representing set membership).

Sets, Graphs, and Decorations

1. 

2. $\{\ [\mathrm{T},\ +\mathrm{Past}],\ \{\ \{\ [\mathrm{n}],\ [\sqrt{\mathit{Harry}}]\ \},\ \{\ [\mathrm{v}],\ [\sqrt{\mathit{Fall}}]\ \}\ \}\ \}$

Omitted is some kind of indexing needed to distinguish different draws from the lexicon.

Sets look OK for individual structures, but for workspaces, we want a ‘unique motherood’ constraint, which seems more natural with graphs.

But . . .

The set representation doesn't work out for the 'core computational structure of Merge' as descrived in sec 3 (pg 20-21), because the terminals are unlabelled, and would lack any indices, so there would be no difference between $\{x\}$ and $\{xx\}$, etc. In general, I'm not at all convinced that this section adds anything to actual understanding beyond what good undergraduate syntax students in GG courses would pick up from the 1970s textbooks.

Note that the text here says nothing about sets, but only 'balanced bracketted expresions'.

What are Trees Made of

1. MCB (pg 2) says 'lexical items' and 'features'
2. MBC (pg 18, 1st sentence of 3.3) says 'lexical items and syntactic objects'. I reckon that this is just a typo.
3. I think that the MCB statement makes for a system with a significant resemblance to LFG, where, if you formulate f-structures as graphs (Kaplan likes sets, I prefer graphs, see Kuhn (2003) for a formalization), the terminals are either ordinary features or PRED-features, which have special properties. Even moreso with the addition of 'traces', to come.
4. To make semantic interpretation work properly, 'lexical items'/PRED-features cannot be the locus of meaning (cf MWE's). My theory of what is going on is Andrews (2008, 2019), and I conjecture that it can be adapted to Hopf Algebraic Minimalism (in final section)

Workspaces

Why are the even there at all? This is not explained in the papers, anywhere afaik.

To my mind, they greatly resemble the premise sets in LFG's glue semantics (basically, intuitionistic implicational linear logic, with various possible elaborations), but in the glue case, the premise set is justified by the fact that the syntactic structure is responsible for introducing the meaning-contributions, and constraining their combination, but not determining it.

I don't see what the workspace concept actually does for a theory of syntactic composition (have some incohate guesses, want an actual explanation). Otoh it does not appear to do any damage.

Accessible Terms (1)

These seem to me to be a bit of a mess.

On pg2 of MCB they are defined as 'the proper nonempty subsets' of a syntactic object *SO*. But since a syntactic object is just a set $\{\alpha, \beta\}$, its proper nonempty subsets are just the singleton sets $\{\alpha\}, \{\beta\}$, so why bother with this concept at all, instead of just talking about 'members' of the syntactic object?

But then: "Merge acting on workspaces consists of a collection of operations $M = \{MA\}$, parameterized by sets A consisting of two syntactic objects α, β. These operations have as input a workspace and produce as output a new workspace, by searching for accessible terms in the given workspace matching the selected objects α, β, producing a new object in the workspace obtained by

Accessible Terms (2)

applying binary set formation, and cancelling the remaining deeper copies of the accessible terms used." The concept wanted here seems to be 'member of*' a syntactic object, since EM combines toplevel *SO*s, and IM can pull out reasonably deeply embedded subtrees.

But on pg 5, (see extract pdf) the confusion deepens; now the 'accessible terms' appear to be sets of terminals, which play no role in the sequel, afaik.

A more viable concept appears in MBC, pg 18 (see extract pdf), where Acc' is given two definitions, but the text specifies that *Acc* is the set of trees (not of nodes, and not of sets of terminals) whose mothers are internal, and Acc' is those plus the top-level ones.

Bialgebra Stuff Starts

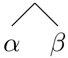
On pg 6, we are told a bit about the product and coproduct, but also that the algebra is going to be a $\mathbb{Z}$-module, but, later on, it is really going to have to be a $\mathbb{Q}$-vector space (to manage the weightings to make Minimal Search minimal).

Next comes the removal of subtrees in the coproduct. The physics way is to contract the subtree to a single point, but MCB reject this for linguistics, on the basis that this subtree would be unlabelled, given the New Minimalism theme that labelling doesn't exist as such, the phenomena being accounted for by Search amongst the terminals (some discussion of this in the new paper with the semantics).

Instead, what is proposed for the quotient T/T_v is to remove the subtree T_v, and then find the unique rooted binary tree that can be obtained from this by contraction of edges.

Oops

But I think I see a problem. Suppose the tree is:



and the subforest F_v of Lemma 2.6 is $\alpha \sqcup \beta$. What is left behind, to appear on the right of the tensor? In the standard physics treatment, it would be:

A reasonable intepretation of contraction would be to combine all three contentless nodes into one, producing the term $(\alpha \sqcup \beta) \otimes \cdot$. But this would lead us with a contentless terminal node of the kind that is not supposed to be produced by Merge. Putting that issue on hold, what happens with Foissy's Lemma 1?

Foissy Lemma 1

This says:

For all $x \in \mathcal{H}_{\mathcal{R}}$ (x a forest),

$$\Delta \circ B^{+}(x) = B^{+}(x) \otimes 1 + (\mathbf{Id} \otimes B^{+}) \circ \Delta(x)$$

Now, if $x = [\alpha\beta]$, then

$$\Delta(x) = x \otimes 1 + 1 \otimes x + \alpha \otimes \beta + \beta \otimes \alpha + (\alpha \sqcup \beta) \otimes \cdot$$

Applying $\mathbf{Id} \otimes B^{+}$ to this, we get:

$$x \otimes \cdot + 1 \otimes [x] + \alpha \otimes [\beta] + \beta \otimes [\alpha] + (\alpha \sqcup \beta) \otimes [\cdot]$$

On the other hand, $B^{+}(x) = [x] = [[\alpha\,\beta]]$, so:

$$\Delta \circ B^{+}(x) = [x] \otimes 1 + 1 \otimes [x] + x \otimes \cdot + \alpha \otimes \beta + \beta \otimes \alpha + (\alpha \sqcup \beta) \otimes \cdot$$

The last three terms are failing to match up.

Coassociativity

It might be possible to fix this by clarifying or changing the cleanup rules, but I think that is beyond my present job description, and is furthermore or questionable usefulness because in the latest paper, bottom of page 5, (MCBsem, I'll call it), they propose to eliminate cleanup so that there will be traces to help with semantics. This would allow us to use Foissy's coassociativity proof, which seemed like a straightforward slog to me. Otoh I have so far not been able to make sense of the one in MCB:7.

And a final observation. At the end of 2.2 (pg 8), MCB observe that limiting Δ to extracting at most one subtree will break coassociativity (which requires Δ to be a homomorphism for the algebra), but I suggest that it might be possible to restrict Δ to removing one subtree from each tree. This would provide for the only linguistically relevant case, and neutralize the fuss about $[\alpha\,\beta]$.

Matching & Merge

Merge is defined with the aid of a notion of ‘matching’, which is never actually defined, but might be something like isomorphism up to possibly different indexes.

For some unknown reason, Merge is defined in terms of ‘matching’ rather than isomorphism. Perhaps because ‘isomorphism’ assumes a graph representation, while ‘matching’ is trying to apply to sets? Idk.

But then, there is a problem with defs defs (2.17) and (2.18), shown on 2nd page of excerpts, which seem to be messed up because of excessively narrow existential quantifier scope in the former. As stated, for any pair (F_1, F_2), there is an F as specified, for example $F_1 \sqcup F_2$, taking the total cut for all the trees of F_1 (and that is not the only way). Further on, F appears out of scope of its existential quantifier.

Matching & Merge 2

But I think if we parameterize δ by F, things can be made to work out. We start by defining:

$$\delta^{F}_{S,S'} : \mathcal{V}(\mathfrak{F}_{\mathcal{SO}_o}) \otimes \mathcal{V}(\mathfrak{F}_{\mathcal{SO}_0}) \to \mathcal{V}(\mathfrak{F}_{\mathcal{SO}_o}) \otimes \mathcal{V}(\mathfrak{F}_{\mathcal{SO}_0})$$

by means of the set:

$$\mathfrak{F}^{\Delta,F}_{\mathcal{SO}_0} = \{(F_1, F_2) \in \mathfrak{F}_{\mathcal{SO}_0} \times \mathfrak{F}_{\mathcal{SO}_0} \mid F_{\underline{v}} \subset F, F_1 = F_{\underline{v}} \;\text{ and }\; F_2 = F/F_{\underline{v}}\}$$

There are then three cases applying to $F_1, F_2 \in \mathfrak{F}_{\mathcal{SO}_0}$.

Matching & Merge 3

1. If (F_1, F_2) aren't in $\mathfrak{F}^{\Delta,F}_{\mathcal{SO}_0}$, then $\mathfrak{F}^{\Delta,F}_{\mathcal{SO}_0}(F_1, F_2) = 0$
2. The next case is when we find subtrees that 'match' S, S', contained inside (full, not sub-)trees T_a, T_b, respectively. If I'm interpreting the double subscripting correctly, these subtrees are T_{a,v_a}, $T_{b.v_b}$, respectively. Then the output is a tensor whose left component is the union S, S' (why not the matching subtrees? Perhaps this is a mistake? (If this was being done by Prolog unification, matching would create identity, so the choice wouldn't matter.) And the right hand component of the tensor is the union/product of three things:

   a T_a, T_b quotiented by S, S' repsectively (or maybe, again, T_{a,v_a}, $T_{b.v_b}$).
   b F with T_a, T_b removed (denoted $F^{(a,b)}$)

And, finally, if there is no match, then the result is $1 \otimes F$.

Merge

Finally, on pg 10, after a brief excursion about B^+, we get to a definition of Merge, repeated in the excerpts, which I think works.

Random observation: somebody with a modest linguistic formalization background would probably approach this by writing set-valued functions that produced multiple outputs from the input (eg a forest), as might be relatively easily implemented in PROLOG (ignoring issues of efficiency); I wonder if physics math is very well adapted to expressing this kind of thing.

So far, the tensors are not doing anything that ordered pairs won't do, and the 'formal sums' are just collections of alternatives, and we have no interest (so far at least) in actually adding up anything derived from them. But, we will see that this approach would also have to provide some desireability scoring for the alternatives.

In Multiple Forms

This formulation admits a variety of different cases of merge, enumerated in 2.4.1, of which we want only two, with how to get them the subject matter of 2.4.2-3.

A superficially confusing feature might be that the ϵ, δ in square brackets in the top line of (2.25) (bottom line of p12) don't do anything: the results of the modified coproduct, as specified by the second line, fall into the codomain if the ordinary one. The sentence "*with this simple bookkeeping device* ... also seems wrong, because so far we just have two parameters, which are going to be raised to powers by properties of the cut in the coproduct, but with no relationship between them.

Minimal Search, 'Best Merge'

Happily, This gets fixed in 2.4.3 immediately below (excerpt from pg 13), where we see that Δ is getting parameterized by $(\epsilon, 1/\epsilon)$, where ϵ appears to be the sum of the distances from the root of the extracted subtrees (yet another typo in the sentence right under the formula at the top of 13), and we are interested in what happens when ϵ is small.

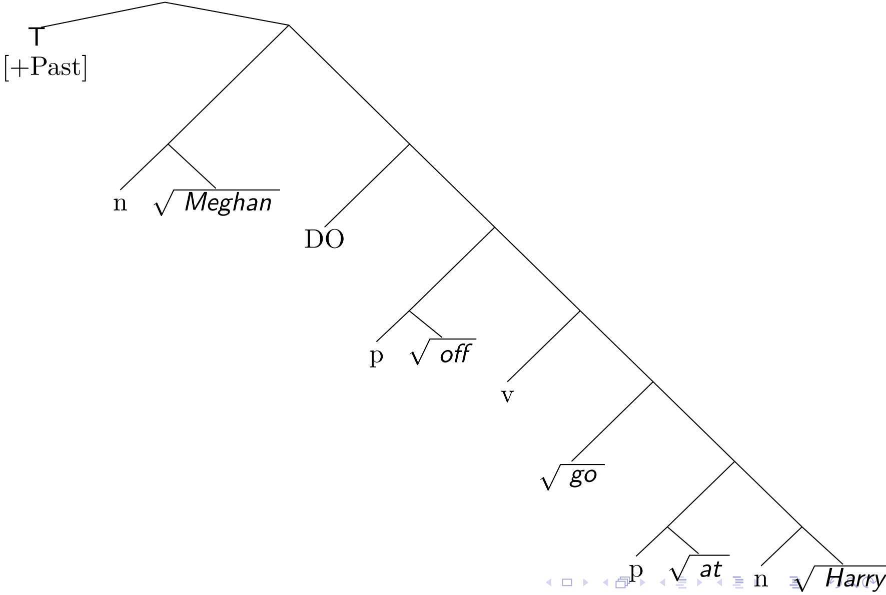
Sadly, the typo hunt ends here, because this is as far as I've gotten.

Perhaps I would have gotten further if I had spent less time trying to come up with an approach to the semantics . . ..

A Note on 'Traces'

The possible reinstatement of 'traces' in MCBsem makes this version of Minimalism a bit more like LFG.

1. LFG f-structures are 'almost acyclic' rooted graphs, with labels on the terminals and arcs, but not on the nodes.
2. MCB trees are rooted acyclic graphs with a binarity restriction, and labels only on the terminals
3. The binarity restriction means that labels aren't needed to provide different structural relations between an item and nearby items that might be its 'arguments'
4. With the reinstatement of traces, there might be something like the limited reentrancies of LFG
5. The limitations on LFG reentrances are very similar to those on IM, but lack anything like a deeper explanation or motivation.
6. But 'adjuncts' make a bit of a mess for everybody.


T
[+Past]
n
$\sqrt{Meghan}$
DO
p
$\sqrt{off}$
v
$\sqrt{go}$
p
$\sqrt{at}$
n
$\sqrt{Harry}$

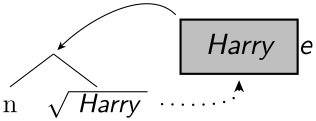
Co-description, MWEs, & Description by Analysis

1. In the scheme of Andrews (2008, 2019), meaning-contributions are introduced by 'semantic lexicon entries' (SLEs), which look at the syntactic structure and introduce contributions, which are then assembled according to rules.
2. In LFG terms, this is 'Description by Analysis', whereas the mainstream view is 'co-description' (Halvorsen & Kaplan, 1995), where the semantic information is introduced inside fairly conventional (lexicalist) lexical entries, along with everything else.
3. I think co-description makes a mess out of multi-word expressions (MWEs), which you can make progress on resolving either my way or Jamie Findlay's way Findlay (2019), using TAGs instead of LFG's annotated c-structure rules.

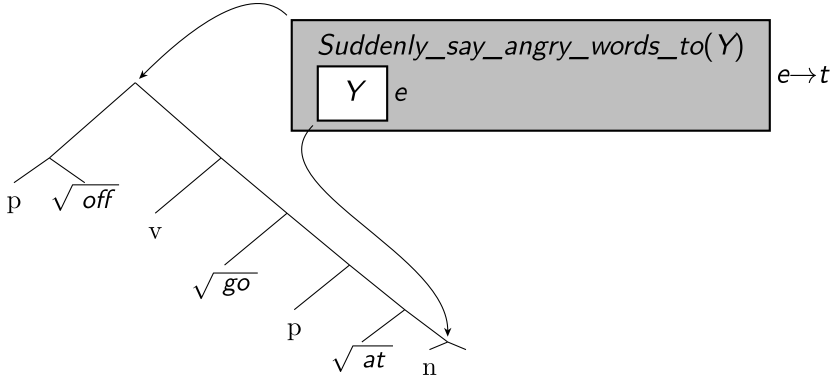
Co-description, MWEs, & Description by Analysis 2

1. SLEs cause certain features, the 'interpretable' ones, to be 'checked off', so that they don't get interpreted again, while others, the uninterpretable ones serve as contextual constraints.
2. The latter seem to be pretty much what is intended by Harley (2014), the former seem absent from that system.

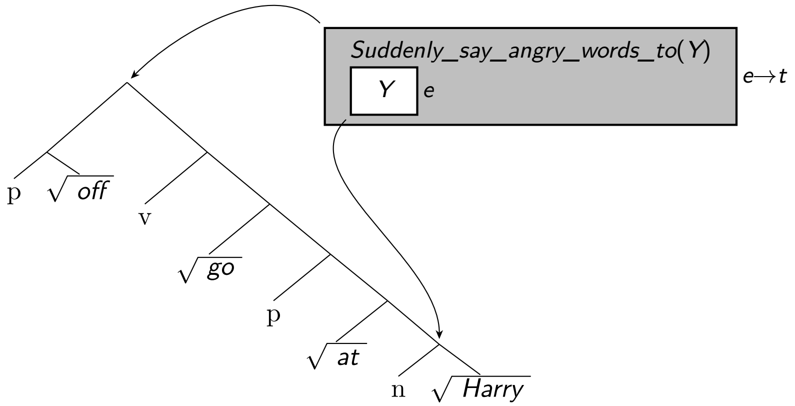
Harry

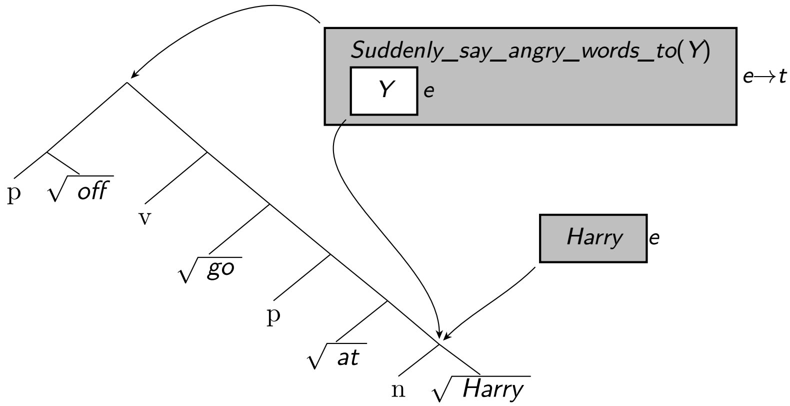


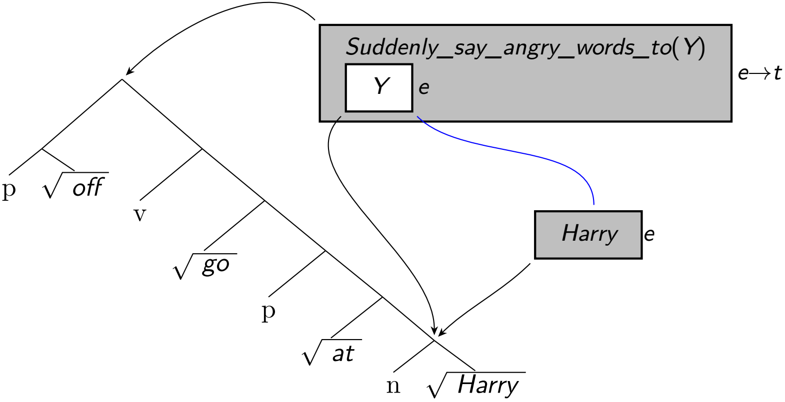
The dotted line represents the checking off of the interpretable root



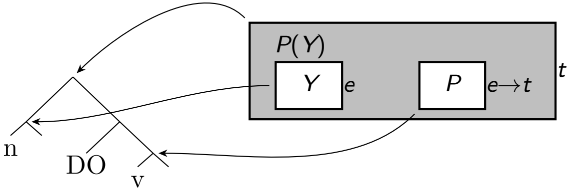
Checking off lines from the 3 roots omitted for less clutter. They would all go to the big shaded box (a function from interpretable features to (sometimes sets of) meaning-contributions).


Suddenly_say_angry_words_to(Y)
Y
e
e→t
p
√off
v
√go
p
√at
n
√Harry


Suddenly_say_angry_words_to(Y)
Y
e
e→t
Harry
e
p
√off
v
√go
p
√at
n
√Harry

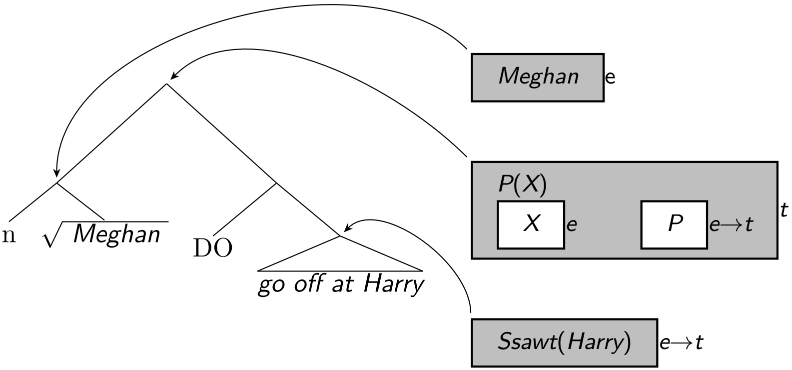


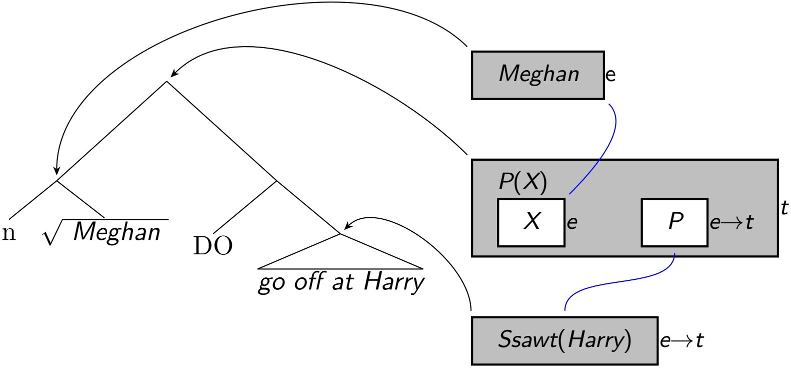
The blue line is a proof-net ‘axiom-link’, along which the substantive values unify with the upper case variables. They don’t have to be interpreted directionally, since they always connect shaded to unshaded boxes (intuitionistic proof-net polarity, c.f. de Groote (1999)). This scheme is a notational variant of Perrier (1999), inspired by Construction Grammar, and hopefully more readable by linguists that the usual ones.



And here is a treatment, almost randomly chosen by me, of the 'external argument'. In proof-nets, it is not usual to have links connecting non-atomic types, and this could be eliminated here, but would make the diagrams more complicated (in ways that would be necessary if we wanted to treat inverse scope of quantifiers, and various other things (requiring actual lambda-calculus, rather than only PROLOG-style substitutional variables).

DO is to be interpreted, the category features uninterpreted (as always; I am taking them to be purely contextural).


Meghan
e
P(X)
X
e
P
e→t
t
Ssawt(Harry)
e→t
n
√ Meghan
DO
go off at Harry



Final result: *Ssawt*(*Harry*)(*Meghan*)

Final Remarks on the Semantics

1. What we have seen is a re-notation of linear intuitionistic proof nets with only Implication Elimination aka Modus Ponens. For quantifiers etc, we need to add Implication Introduction aka Hypothetical Deduction.
2. The basic ideas are from de Groote and Perrier, as cited above
3. The notation is from Kay & Fillmore (1999), who appear to have invented the most rudimentary form of linear logic without knowing it, plus a topological intepreration.
4. There is an extension, not discussed here, to manage lambda-binding a.k.a Implication Introduction, also with a topological interpretation in terms of things fitting inside each other.
5. But the topological interpretation is not obvious and likely nonexistent for some common extensions of the glue system, such as tensors and monads.

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.

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

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. 1995. Projections and semantic description in Lexical-Functional Grammar. In Mary Dalrymple, Ronald M. Kaplan, John T. Maxwell & Annie

Zaenen (eds.), *Formal issues in Lexical-Functional Grammar*, Stanford, CA: CSLI Publications.

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.

Kuhn, Jonas. 2003. *Optimality-theoretic syntax–a declarative approach*. Stanford CA: CSLI Publications.

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