Glue via Correspondence

Avery Andrews, ANU, Sep 2006

In this note I will show that the universally quantified f-structure variables that have so far been used in LFG glue logic to describe quantifier-scope can be dispensed with, by making greater use of correspondence architecture as employed elsewhere in LFG.[1]

Some conventional although somewhat streamlined meaning-constructors for the sentence *everybody seems to sleep* are:

(1) $Everybody : (g_e \multimap H_t) \multimap H_t$

$Sleep : g_e \multimap f_t$

$Seem : h_t \multimap f_t$

The streamlining consists of removing reference to the 'semantic projection', and having the f-structure variable H be implicitly rather than explicitly universally quantified. Both of these issues will be discussed below.

The atomic formulas, or 'literals',[2] are conventionally seen as consisting entirely of the f-structure and type information they provide (including, possibly, dummy attributes on a semantic projection that comes off of f-structure), and so, since we want the type t literals of the quantifier to associate with either the matrix f-structure f or the complement f-structure h (but to the same structure in either case), there needs to be a variable of some kind, and the linear universal quantifier is the appropriate binder.

Suppose, however, we thought of the occurrences of the atomic formulas in the meaning-constructors as all being at the outset distinct objects, with certain attributes, which will be put into constrained correspondences with various other things. Most importantly, each atomic formula object is to have a corresponding f-structure, under a correspondence σ from the atomic formula occurrence objects to the substructures of the f-structures. Like the c-structure-to-f-structure correspondence ϕ, σ is a function, but is in general neither one-to-one (because of quantifiers and 'head-switching' modifiers) nor onto (because of expletives). The f-structure labels and subscripts in the conventional formulations of (1) then specify information about these correspondents. In addition to σ, we need a projection or attribute τ for the semantic type information, which so far can be treated as a function into a rather small set of objects, representing basic types in a simple type theory. The resulting meaning of the *Everybody* constructor above can be stated more explicitly as follows:

(2) $everybody : (X \Rightarrow Y) \Rightarrow Z$

[1] For recent discussion and background on these variables, see Kokkonidis (2006), for correspondence architecture, Kaplan (1995).

[2] So-called because in a formulation of the logic based on tensor, par and negation, some of them would be negations of atomic formulas.

$$\sigma(X) = g \qquad \tau(X) = e$$
$$\sigma(Y) = \sigma(Z) \qquad \tau(Y) = t$$
$$\tau(Z) = t$$

Observe that the variable has been removed from the type t positions, and replaced by an equation. Another way of thinking about it is that all of the atomic formulas are variables, but only some of them are equated to each other, and the correspondence machinery rather than the assembly logic is responsible for binding them. Missing from this account is the 'semantic projection' of standard glue, since it doesn't appear to have acquired any empirically useful function since it was originally proposed.[3]

If the atomic formulas in the meaning-constructors remain distinct, no conclusion can be derived, so the next step is group them into pairs, amounting to the addition of axiom-links in a proof-net. The pairing is done subject to the following conditions, plus one more, to come:

(3) a. Only nonoverlapping pairs of opposite polarity are paired (see Fry (1999), Andrews (2004), or the general proof-net literature for discussion of polarity).

b. Only one literal of negative polarity is left unpaired (in the standard polarity assignment, which Andrews (2004) reverses, so that the unmatched polarity is positive[4]).

c. Paired items have the same f-structure and type correspondents.

At the level of the formulas, the pairing is to result in the paired atomic formulas becoming identified, or merged, so that it becomes possible to derive some conclusions from the meaning-constructors. In conventional formulations of proof-nets, the formulas connected by axiom-links are considered to be distinct objects, labelled by the same formula, but for our purposes, treating them as being the same object as the pleasant consequence that the agreement requirement (c) above becomes simply the requirement that the σ and τ correspondences remain single-value under the mergings.

The two possible mergings for the constructors of (1) are given here, where I've installed superscripts for the type information and subscripts for the f-structure locations:

(4) a. *Everybody* : $(X^e_g \multimap W^t_f) \multimap Z^t_f$

Sleep : $X^e_g \multimap Y^t_h$

Seem : $Y^t_h \multimap W^t_f$

[3] Apparently on grounds that were largely esthetic, to keep certain 'dummy attributes' out of the f-structure (Kokkonidis 2006). However the dummy attributes don't actually appear to do anything either, so it would seem best to just eliminate the projection as originally conceived (although one could think of the present proposal as a radical reconstrual of it, in which the direction is reversed and the relata different).

[4] Andrews' reversed polarities seem more natural for thinking about semantic assembly in languages, although less in accord with the logical interpretation of the proof-nets. Here we use the standard polarities for the sake of greater congruence with the proof-net literature.

b. *Everybody* : $(X^e_g \multimap W^t_f) \multimap Y^t_f$
Sleep : $X^e_g \multimap {}^t_f W$
Seem : $Y^t_f \multimap Z^t_f$

The upper-case letters are the atomic formulas here, while the subscripts and superscripts are just extra information about the correspondence relations that constrained the pairings. Perhaps the proposition letters og (4) should be in bold font to indicate that no further mergers are allowed.

A set of formulas produced in accordance with (3) is equivalent to an intuitionistic implicational proof-structure (tensors for the full multiplicative fragment could be added without creating significant issues) as defined in the proof-net literature, but, unfortunately, not every proof-structure corresponds to a valid proof, or to a sensible reading for a sentence.

So we need to apply one of the many known versions of the correctness criterion, to see if the proof-structure is a proof-net, or, equivalently, if we can prove the unmatched atomic formula from the meaning-constructors in Linear Logic. Pairings for which this can't be done, or which, equivalently, fail the correctness criterion, are rejected as ill-formed. The term-labelling on a deduction will produce a conventional-looking logical form, or one can be produced from the proof-net itself by a method such as that of Perrier (1999), although deGroote and Retoré (1996) shows that we don't really have to do this: the proof-net itself is perfectly adequate as a 'logical form'.

The result is that putting together a semantic structure with glue can be conceptualized so as to be very similar to putting together a c-structure from the partial trees provided by c-structure rules, a matter of matching inputs and outputs, subject to constraints. The difference, in the standard conception, is that c-structure is generated freely, imposing constraints on the other levels, which then filter the results of the c-structure generation, whereas the initial assembly of the meaning-constructors is heavily constrained by relationships to the already-extant level of f-structure. But it's possible to set up architectures such as that of Andrews (to appear) where this is not the case, and the generation of c-structures and semantic structures is on an equal footing, but subject to the constraint that the resulting f-structures match closely enough.[5]

I will make two further observations, one on the nature of the required assembly logic, and the other on the 'semantic projection'. As a consequence of the construction of the formula-set, with 'sparse' distribution of atomic formulas amongst the positions in the assumptions of the deduction/proof-net, there are relatively few things that can be proved from them. Indeed, even in Classical Logic, there will be only one way to derive the wanted final conclusion, and this will require using all of the assumptions. This construal of assembly thus provides an opportunity to explore whether there might be any useful results from Kokkonidis' 2006 observation that the assembly logic might not have to be linear after all. *Prima facie*, it seems to be at least *de facto* Relevant

[5]This shows that Pullum and Scholz's (2001) distinction between 'Model Theoretic' and 'Generative-Enumerative' approaches to syntactic theory is a bit simplistic, especially applied to LFG, which uses both kinds of technique.

(no Weakening/vacuous binding), but it is not 100% clear that Contraction/multiple binding might not be useful in certain areas, such as for bound anaphora. Presumably this could be implemented by allowing one negative to merge with several positives; note that most potential instances of this would be blocked by the f-structural correspondence requirements.

As a final observation, I'll say something about the standard semantic projection. Its main use has been as a place to put the dummy attributes VAR and RESTR, so that quantificational determiner and common noun meaning-constructors can be stated like this:

(5) $$\begin{aligned} \textit{Every} &: ((\uparrow_\sigma \text{VAR}))_e \multimap (\uparrow_\sigma \text{RESTR})_t) \multimap (\uparrow_{\sigma e} \multimap H_t) \multimap H_t \\ \textit{Dog} &: (\uparrow_\sigma \text{VAR})_e \multimap (\uparrow_\sigma \text{RESTR})_t \end{aligned}$$

The function of the dummy attributes would have been originally to provide different locations for the calculation of the predicate provided for the whole NP and that of the containing proposition, but there is no reason why such different locations have to be set up. If the constructors are just:

(6) $$\begin{aligned} \textit{Every} &: (\uparrow_e \multimap \uparrow_t) \multimap (\uparrow_e \multimap H_t) \multimap H_t \\ \textit{Dog} &: (\uparrow_e \multimap \uparrow_t) \end{aligned}$$

nothing goes wrong, basically because in any well-formed structure, the variable effectively introduced by the first $\uparrow_e$ of the quantifier constructor needs to be bound/discharged when the meaning corresponding to the $(\uparrow_e \multimap \uparrow_t)$-argument is calculated.

If some clear use for dummy attributes and a semantic projection[6] was found, they could be constructed by the meaning-constructors with formulations such as $(\sigma(X)_\delta\, \text{VAR})$, where δ is a projection of f-structure for holding dummy attributes. But there is currently no justification for positing such things.

Bibliography

Andrews, A. D. 2004. Glue logic vs. spreading architecture in LFG. In C. Mostovsky (Ed.), *Proceedings of the 2003 Conference of the Australian Linguistics Society*. URL: `http://http://www.als.asn.au/`.

Andrews, A. D. to appear. Generating the input in OT-LFG. In Grimshaw, Maling, Manning, and Zaenen (Eds.), *Architectures, Rules, and Preferences: A Festschrift for Joan Bresnan*. Stanford CA: CSLI Publications. URL: `http://arts.anu.edu.au/linguistics/People/AveryAndrews/Papers`.

[6]From the perspective of Andrews and Manning (1999), sufficient justification for a projection distinct from f-structure would be that the sharing behavior be different. Somewhat intriguingly, Andrews and Manning (1999) and Andrews (2004) appear to jointly suggest that 'propositional' information spreads less aggressively than f-structure, motivating Andrews and Manning's concept of 'argument structure', while 'referential' information spreads more aggressively, so that various levels of attributive modification in the NP all seem in essence to describe the same entity.

Andrews, A. D., and C. D. Manning. 1999. *Complex Predicates and Information Spreading in LFG*. Stanford, CA: CSLI Publications.

Dalrymple, M., R. M. Kaplan, J. T. Maxwell, and A. Zaenen (Eds.). 1995. *Formal Issues in Lexical-Functional Grammar*. Stanford CA: The Center for the Study of Language and Information.

deGroote, P., and C. Retoré. 1996. On the semantic reading of proof-nets. In G. G.-J. Kruijff and D. Oehrle (Eds.), *Formal Grammar*, 57–70, FOLLI Prague, August. URL: `citeseer.ist.psu.edu/degroote96semantic.html"`.

Fry, J. 1999. Proof nets and negative polarity licensing. In M. Dalrymple (Ed.), *Syntax and Semantics in Lexical Functional Grammar: The Resource-Logic Approach*, 91–116.

Kaplan, R. M. 1995. The formal architecture of LFG. In M. Dalrymple, R. M. Kaplan, J. T. Maxwell, and A. Zaenen (Eds.), *Formal Issues in Lexical-Functional Grammar*, 7–27. CSLI Publications.

Kokkonidis, M. 2006. First order glue. *Journal of Logic, Language and Information*. to appear; available on request according to `http://users.ox.ac.uk/~lina1301/`.

Perrier, G. 1999. Labelled proof-nets for the syntax and semantics of natural languages. *L.G. of the IGPL* 7:629–655. URL: `http://www.loria.fr/~perrier/papers.html`.

Pullum, G. K., and B. C. Scholz. 2001. On the distinction between model-theoretic and generative-enumerative syntactic frameworks. In *LACL*, 17–43. URL: `http://link.springer.de/link/service/series/0558/bibs/2099/20990017.htm`.