Categorial Grammar and Lexical-Functional Grammar[1]

In this paper we discuss a version of categorial grammar that can also be interpreted as a version of LFG. It is obvious that a convergence between the two frameworks has set in since [3] proposed to use Linear Logic as a 'glue' between f-structure and semantics, as Intuitionistic Linear Logic (**ILL**) is nothing but an extension of the undirected Lambek Calculus. The present proposal is meant to bring this convergence one step further.

Two simplifications of the global architecture of LFG seem possible. Firstly, semantics is read off from f-structure with the help of derivations in what is largely the $\{\multimap, \otimes\}$-fragment of **ILL**. With some currying, much of the work can be done with just derivations in the $\multimap$-fragment. But, by an observation in [1], these derivations are isomorphic to the *single bind combinators* via the usual Curry-Howard isomorphism. (*Combinators* ([2]) are closed λ-terms built up from variables with the help of λ-abstraction and application only. A λ-term is *single bind* if each λ binds exactly one free variable.) This suggests that single bind combinators can take the place of glue derivations. The second simplification I think possible is a replacement of LFG's phrase structure component by the combinator / linear logic engine. LFG currently has two 'engines' providing generative power, phrase structure rules and linear logic. It is simpler to have just one.

In order to make what I have in mind clearer, there is a mini-grammar on the example sheet. The items in (1) form its lexicon. Each is a *sign* consisting of three parts: 1) a λ-term describing c-structure, 2) a λ-term embodying its semantics, and 3) a λ-term over the language defined in [4], describing f-structure. The idea is that signs can freely be combined using single bind combinators. In (2) three examples of the latter are given. The mode of combination is *pointwise* (compare [5]): The result of pointwise application of a combinator $\mathcal{C}$ to signs $S_1, \ldots, S_n$ is defined as in (3), where S^i is the i-th component of S and the $\mathcal{C}^i$ are suitably typed versions of $\mathcal{C}$ (for further information on typing and applicability see the full paper).

For example, consider the result of pointwise applying (2a) to (1b), (1c), and (1a) (in that order). A series of lambda conversions will show that (4) is obtained. This then, is a generated sign; it is *admissable* since its third component is non-empty in some model of the first four axioms in [4]. For the other scope possibility, try pointwise application of (2b) to (1b), (1c) and (1a). Note that if (2c) is used to combine these signs, still in the same order, a permutation of argument places will result. But since syntax and semantics permute in tandem, no harm is done.

The system has obvious affinities not only with LFG and Lambek Categorial Grammar, but also with Combinatory Categorial Grammar (see e.g. [6]), although it is non-directional and does not use derivations.

[1] Submitted for a talk at LFG2001, Hongkong.

Example Sheet

(1) a. $\langle \lambda T.T(\mathsf{every} \bullet \mathsf{man}),$
$\lambda P \forall x[man(x) \rightarrow P(x)],$
$\lambda \mathcal{F}.\mathcal{F}(\lambda f.arc(f, cat, N) \wedge arc(f, num, sg) \wedge arc(f, pers, 3))\rangle$

b. $\langle \lambda T.T(\mathsf{a} \bullet \mathsf{woman}),$
$\lambda P \exists x[woman(x) \wedge P(x)],$
$\lambda \mathcal{F}.\mathcal{F}(\lambda f.arc(f, cat, N) \wedge arc(f, num, sg) \wedge arc(f, pers, 3))\rangle$

c. $\langle \lambda t_1 \lambda t_2.(t_2 \bullet (\mathsf{loves} \bullet t_1)),$
$\lambda x \lambda y.love(y, x),$
$\lambda F_1 F_2 \lambda f \exists f_1 f_2[F_1(f_1) \wedge F_2(f_2) \wedge arc(f, cat, V) \wedge arc(f, tense, pres) \wedge$
$arc(f_1, cat, N) \wedge arc(f, obj, f_1) \wedge arc(f_2, cat, N) \wedge arc(f_2, num, sg) \wedge$
$arc(f_2, pers, 3) \wedge arc(f, subj, f_2)]\rangle$

(2) a. $\lambda \mathcal{Q}_1 \lambda \mathcal{R} \lambda \mathcal{Q}_2.\mathcal{Q}_2(\lambda z_2.\mathcal{Q}_1(\lambda z_1.\mathcal{R}(z_1)(z_2)))$

b. $\lambda \mathcal{Q}_1 \lambda \mathcal{R} \lambda \mathcal{Q}_2.\mathcal{Q}_1(\lambda z_1.\mathcal{Q}_2(\lambda z_2.\mathcal{R}(z_1)(z_2)))$

c. $\lambda \mathcal{Q}_1 \lambda \mathcal{R} \lambda \mathcal{Q}_2.\mathcal{Q}_2(\lambda z_2.\mathcal{Q}_1(\lambda z_1.\mathcal{R}(z_2)(z_1)))$

(3) $\langle \mathcal{C}^1(S^1_1) \cdots (S^1_n), \mathcal{C}^2(S^2_1) \cdots (S^2_n), \mathcal{C}^3(S^3_1) \cdots (S^3_n)\rangle$

(4) $\langle((\mathsf{every} \bullet \mathsf{man}) \bullet (\mathsf{loves} \bullet (\mathsf{a} \bullet \mathsf{woman}))),$
$\forall x[man(x) \rightarrow \exists y[woman(y) \wedge love(x, y)]],$
$\lambda f \exists f_1 f_2[arc(f, cat, V) \wedge arc(f, tense, pres) \wedge arc(f_1, cat, N) \wedge arc(f, obj, f_1) \wedge$
$arc(f_2, cat, N) \wedge arc(f_2, num, sg) \wedge arc(f_2, pers, 3) \wedge arc(f, subj, f_2)]\rangle$

References

[1] J.F.A.K. van Benthem. *Essays in Logical Semantics.* Reidel, Dordrecht, 1986.

[2] H. Curry and R. Feys. *Combinatory Logic*, volume I. North-Holland, Amsterdam, 1958.

[3] M. Dalrymple, J. Lamping, and V. Saraswat. LFG Semantics via Constraints. In *Proceedings of the Sixth Meeting of the European ACL.* European Chapter of the Association for Computational Linguistics, 1993.

[4] M. Johnson. Logic and Feature Structures. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, Sydney, Australia, 1991.

[5] R.T. Oehrle. Multi-Dimensional Compositional Functions as a Basis for Grammatical Analysis. In R.T. Oehrle, E. Bach, and D. Wheeler, editors, *Categorial Grammars and Natural Language Structures*, pages 349–389. Reidel, Dordrecht, 1988.

[6] M. Steedman. *Surface Structure and Interpretation.* MIT Press, 1996.