To cite this version : Retoré, Christian *Type-theoretical natural language semantics: on the system F for meaning assembly*. (2013) In: Conference of the Types Project - TYPES 2013, 22 April 2013 - 26 April 2013 (Toulouse, France).

Type-theoretical natural language semantics: on the system F for meaning assembly

Christian Retoré*

Équipe Melodi, IRIT-CNRS, UPS, 118 route de Narbonne, 31062 Toulouse Cedex 9
(& LaBRI, Université de Bordeaux, 351 cours de la Libération, 33405 Talence cedex)

Roughly speaking, the standard semantical analysis of natural language consists in mapping a sentence $s = w_1 \cdots w_n$ to a logical formula $[\![s]\!]$ which depicts its meaning. It is a computational process which implements Frege's compositionally principle. A parser turns the sentence s into a binary parse tree t_s specifying at each node which subtree is the function — the other subtree being its argument. The lexicon provides each leaf of t_s, that is a word w_i, with a λ-term $[\![w_i]\!]$ over the base types t (propositions) and e (individuals). By structural induction on t_s, we obtain a λ-term $[s]$:t corresponding to t_s. Its normal form, that is a formula of higher order logic, is $[\![s]\!]$:t, the meaning of s. This standard process at the heart of Montague semantics relies on Church's representation of formulae as simply typed λ-terms, see e.g [7, Chapter 3].

It would be more accurate to have many individual base types rather than just e. This way, the application of a predicate to an argument may only happen when it makes sense. For instance sentences like *"The chair barks."* or *"Their five is running."* are easily ruled out when there are several types for individuals by saying that *"barks"* and *"is running"* apply to individuals of type *"animal"*. Nevertheless, such a type system needs to incorporate some flexibility. Indeed, in the context of a football match, the second sentence makes sense, because *"their five"* may be understood as a player who, being *"human"*, is an *"animal"* that can run.

These meaning transfers have been receiving much attention since the 80s, as [1] shows. As [1, 5], we too proposed a formal and computational account of these phenomena, based on Girard's system F (1971) [3]. We explored the compositional properties (quantifiers, plurals and generic elements,....) as well as the lexical issues (meaning transfers, copredication, fictive motion,...) [2, 8, 9]. Our system works as follows: the lexicon provides each word with a main λ-term, the "usual one" which specifies the argument structure of the word, by using refined types: *"runs:* $\lambda x^{animal}\underline{run}(x)$*"* only applies to *"animal"* individuals. In addition, the lexicon may endow each word with a finite number of λ-terms (possibly none) that implement meaning transfers. For instance a *"town"* may be turned into an *"institution"*, a geographic *"place"*, or a football *"club"* by the optional λ-terms *"*f_i*: town*$\rightarrow$*institution"*, *"*f_p*: town* $\rightarrow$ *place"* and *"*f_c*: town* $\rightarrow$ *club"* — in subtler cases these λ-terms may be more complex than simple constants. Thus, a sentence like *"Liverpool is a large harbour and decided to build new docks."* can be properly analysed. Some meaning transfers, like f_c, are declared to be *rigid* in the lexicon. Rigidity prohibits the simultaneous use of other meaning transfers. For instance, the rigidity of f_c properly blocks *"* Liverpool defeated Chelsea and decided to build new docks."*.

The polymorphism of system F is a welcome simplification. For instance, a single type $\Pi\alpha.(\alpha \rightarrow \mathsf{t}) \rightarrow \mathsf{t}$ is enough for the quantifiers $\forall$ or $\exists$x. Polymorphism also allows a factorised treatment of conjunction for copredication: *whenever* an object x of type ξ can be viewed both as an object of type α to which a property $P^{\alpha\rightarrow\mathsf{t}}$ applies and as an object of type β to which a property $Q^{\beta\rightarrow\mathsf{t}}$ applies (via two optional terms $f_0 : \xi \rightarrow \alpha$ and $g_0 : \xi \rightarrow \beta$), x enjoys $P \wedge Q$ can be expressed with $\Lambda\alpha\Lambda\beta\lambda P^{\alpha\rightarrow\mathsf{t}}\lambda Q^{\beta\rightarrow\mathsf{t}}\Lambda\xi\lambda x^{\xi}\lambda f^{\xi\rightarrow\alpha}\lambda g^{\xi\rightarrow\beta}.\ (\wedge^{\mathsf{t}\rightarrow\mathsf{t}\rightarrow\mathsf{t}}\ (P\ (f\ x))(Q\ (g\ x)))$, i.e., with a *single* polymorphic *"and"*. Our logical system also has two layers that slightly differ from

*This research was supported by the ANR projects Loci and Polymnie.

Montague's. Our *meta logic* (a.k.a. glue logic) is system F (instead of simply typed λ-calculus) with base types $\mathtt{t}$, $(\mathtt{e}_i)_{i\in I}$ (instead of a single type $\mathtt{e}$) Our *logic for semantic representations* is many-sorted higher-order logic — the $\mathtt{e}_i$ being the sorts. Quantifiers are preferably represented by Hilbert's operators, that are constants $\epsilon, \tau : \Lambda\alpha.\ (\alpha \to \mathtt{t}) \to \alpha$ [8]. An easy but important property holds: if the constants define an n-order q-sorted logic, any (η-long) normal λ-term of type $\mathtt{t}$ corresponds to a formula of n-order q-sorted logic (possibly $n = \omega$).

We preferred system F to modern type theories (MTT) of [4] and to the categorical logic of [1] because of its formal simplicity and its absence of variants. Furthermore, F terms with a problematic complexity are avoided, since semantical terms derive from the simple terms in the lexicon by means of simple syntactic rules. Nevertheless there are two properties of [4] that are welcome: a proper notion of subtyping, mathematically safe and linguistically relevant, and predefined inductive types with specific reduction rules. Indeed, subtyping naturally represents ontological inclusions (a *"human being"* is an *"animal"*, hence predicates that apply to *"animals"* also apply to *"human beings"*). Coercive subtyping [11] sounds promising for F. Its key property is that, if at most one subtyping map is given between any two base types, then there also is at most one subtyping map between any two complex types. Predefined (inductive) types, e.g. integers as in Gödel's system T and finite sets of α-objects with their reduction schemes as in [10] are also welcome — encodings in F are cumbersome. The key points are normalisation, confluence and the absence of closed constant-free terms in any false type. We shall also illustrate the linguistic relevance of these extensions, which are already included in Moot's semantical and semantical parser for French named Grail. [6]

References

[1] Nicholas Asher. *Lexical Meaning in context – a web of words.* Cambridge University press, 2011.

[2] Christian Bassac, Bruno Mery, and Christian Retoré. Towards a Type-Theoretical Account of Lexical Semantics. *Journal of Logic Language and Information*, 19(2):229–245, April 2010. `http://hal.inria.fr/inria-00408308/`.

[3] Jean-Yves Girard. *The blind spot – lectures on logic.* European Mathematical Society, 2011.

[4] Zhaohui Luo. Contextual analysis of word meanings in type-theoretical semantics. In Sylvain Pogodalla and Jean-Philippe Prost, editors, *LACL*, volume 6736 of *LNCS*, pages 159–174. Springer, 2011.

[5] Zhaohui Luo. Common nouns as types. In Denis Béchet and Alexander Ja. Dikovsky, editors, *LACL*, volume 7351 of *Lecture Notes in Computer Science*, pages 173–185. Springer, 2012.

[6] Richard Moot. Wide-coverage French syntax and semantics using Grail. In *Proceedings of Traitement Automatique des Langues Naturelles (TALN)*, Montreal, 2010.

[7] Richard Moot and Christian Retoré. *The logic of categorial grammars: a deductive account of natural language syntax and semantics*, volume 6850 of *LNCS*. Springer, 2012. `http://www.springer.com/computer/theoretical+computer+science/book/978-3-642-31554-1`.

[8] Christian Retoré. Variable types for meaning assembly: a logical syntax for generic noun phrases introduced by "most". *Recherches Linguistiques de Vincennes*, 41:83–102, 2012. `http://hal.archives-ouvertes.fr/hal-00677312`.

[9] Christian Retoré. Sémantique des déterminants dans un cadre richement typé. In Emmanuel Morin and Yannick Estève, editors, *Traitement Automatique du Langage Naturel, TALN 2013*, pages 1–14. ACL Anthology, 2013.

[10] Sergei Soloviev and David Chemouil. Some Algebraic Structures in Lambda-Calculus with Inductive Types. In Stefano Berardi, Mario Coppo, and Ferruccio Damiani, editors, *TYPES*, volume 3085 of *Lecture Notes in Computer Science*, pages 338–354. Springer, 2003.

[11] Sergei Soloviev and Zhaohui Luo. Coercion completion and conservativity in coercive subtyping. *Annals of Pure and Applied Logic*, 1-3(113):297–322, 2000.