Specimens: “most of'' generic NPs in a contextually flexible type theory

An example of second order λ-calculus for meaning assembly

Christian Retoré

LaBRI-CNRS & Université de Bordeaux

Genius III — Paris 2011 Dec. 5-6

Part I
The viewpoint, the question(s)

1. Foreword / apologies

New to the field, unaware of existing work on the truth reference of such issues...
... so thanks to

- Sarah-Jane Conrad (Bern)
- the reviewers
- Alda Mari, David Nicolas and Claire Beyssade

2. An iconoclast point of view

Neither a study of truth nor of reference

- too difficult (vagueness)
- other models (i.e. interaction, proofs and refutation) seem to be more adequate

but a study of the **syntax of semantics**.

3. Syntax of semantics, logical syntax

Montague's view:

syntax ⟶ (logical form) ⟶ **truth, reference in possible worlds**

our view:

syntax ⟶ **logical form** (this talk) ⟶ interaction models, proofs and refutations (later on)

4. Why **type theory** for the syntax of semantics

opposed to Frege's single sort view:
$\forall x : A\ P(x) \Longleftrightarrow \forall x.\ A(x) \rightarrow P(x)$
(impossible for "most of")

in ancient and especially medieval philosophy (in particular Abu Barakat, Avicenna):
we assert properties of things as being member of some class (= type?)

There are less types than logical formulae with a single free variable, they are more constrained, and not any formula defines a comparison class.

5. A personal view on the border between semantics and pragmatics

- semantics is encoded by the terms:
  they yield formulae by compositionality
- pragmatics is encoded in the types
  they are flexible and determined by the context

6. Generic NPs

How do we logically formulate "most of" (much more than "the majority of") generic elements

(1) The AKC notes that any dog may bite [...]

(2) The Brits love France.

(3) Un chien, ça peut toujours mordre.

idea to consider a fictive of fake element, like the τ and ε of Hilbert like the ι of van Heusinger.

(actually there is an other reading for 2 that we are starting to think about: more that the Germans, and other similar classes both being in addition to reference)

7. Radical minimalism / contextualism

Once we we appeal to comparison classes (a type) and its generic element we can address the following puzzle issued from Frege's view of a single domain:

- My daughter is tall and thin for a 2 year old.
- My two-year-old can't get his own cup [...] because he can't reach, [...]

Carlotta who is a two year old girl it can be both tall and not tall, depending on her comparison class (her type in our type theoretic framework). Our type theoretical framework provides an account for such phenomena comparing Carlotta to the generic element of the corresponding class.

Part II

Extending the type system: ΛTy_n

Works both for

Montague semantics

or λ-DRT.

8. What is in a standard semantic lexicon?

Usually a λ-term associated with a word which is used according to the syntactic structure.

If multi-sorted ($\mathbf{e}$ is divided into several base types) **type mismatch** is used to block sentence with no (or very unlikely) readings like *"the chair shouts"*.

9. What is in our semantic lexicon?

Here a second order λ-term (see below)depicting the argumental structure and several optional (second order) λ-terms to overpass the type mismatch when licit.

This give rise to transformations from a type to another, with a mechanism for blocking inplausible readings.

10. An example

A "town" can be viewed as some institution(s), as its mayor, as its people, as its location, as its soccer team,...

(4) Liverpool is a poor town and an important harbour.

(5) * Liverpool defeated Chelsea and is an important harbour.

11. Second order types (Girard's F).

Ty_n (several base types) filters the sort of the argument according to lexical constraints, but we need more:

- type variables and quantification over types (flexibility)
- operations that act uniformly upon all types,

Such features exists in Girard system F and can be used for

- **syntax of (generalized) quantification,** here
- plurals with Moot
- facets and co-predication (dot objects) with Bassac & Mery

12. More general types and terms. Second order types (Girard's F).

- Constants $\mathbf{e}$ (or $\mathbf{e}_1, \ldots, \mathbf{e}_n$ in a multisorted system) and $\mathbf{t}$ are types
- any type variable α in a given countable set P is a type.
- Whenever T is a type and α a type variable which may but need not occur in T, $\Pi\alpha.\ T$ is a type.
- Whenever T_1 and T_2 are types, $T_1 \to T_2$ is also a type.

13. More general types and terms. Second order terms (Girard's F).

- A variable x of type T i.e. $x : T$ or x^T is a *term*. Countably many variables of each type.
- A constant k of type T i.e. $k : T$ or k^T is a *term*. Finitely many constant for some types.
- $(f \ \ \tau)$ is a term of type U whenever $\tau : T$ and $f : T \rightarrow U$.
- $\lambda x^T.\ \tau$ is a term of type $T \rightarrow U$ whenever $x : T$, and $\tau : U$.

14. Second order terms (Girard's F): second order rules

- $\tau\{U\}$ is a term of type $T[U/\alpha]$ whenever $\tau : \Lambda\alpha.\ T$, and U is a type.
- $\Lambda\alpha.\tau$ is a term of type $\Pi\alpha.T$ whenever α is a type variable, and $\tau : T$ is without any free occurrence of the type variable α.

15. More general types and terms. Second order reduction.

The reduction is defined as follows:

- $(\Lambda\alpha.\tau)\{U\}$ reduces to $\tau[U/\alpha]$ (remember that α and U are types).
- $(\lambda x.\tau)u$ reduces to $\tau[u/x]$ (usual reduction).

Reduction is strongly normalising and confluent (Girard, 1971): *every term of every type admits a unique normal form which is reached no matter how one proceeds.*

In fact, whenever a term t is of type $\mathbf{t}$ with an appropriate set of constants then its normal form t^0 is a formula (of multi sorted higher order logic).

16. More general types and terms. A second order example.

Given two predicates $P^{\alpha\to t}$ and $Q^{\beta\to t}$
over entities of respective kinds α and β
when we have two morphisms from ξ to α and to β
we can coordinate entities of type ξ:

$$\Lambda\xi\lambda x^{\xi}\lambda f^{\xi\to a}\lambda g^{\xi\to b}.$$
$$(\mathit{and}\ (P\ (f\ x))(Q\ (g\ x)))$$

One can even quantify over the predicates P, Q and the types α, β to which they apply:

$$\Lambda\alpha\Lambda\beta\lambda P^{\alpha\to t}\lambda Q^{\beta\to t}\Lambda\xi\lambda x^{\xi}\lambda f^{\xi\to\alpha}\lambda g^{\xi\to\beta}.$$
$$(\mathrm{and}\ (P\ (f\ x))(Q\ (g\ x)))$$

17. What's new with F ?

- used for the syntax of semantics (a.k.a. meta-logic, glue logic)
- the formulae of semantics are the usual ones
- a single constant, e.g. for the quantifier $\forall$ or the choice function ι which is specialized for each type
- less types (constrained) than formulae with a free variable (e.g. types $\sim$ comparison classes).

It is also the type system of the polymorphic functional programming languages ML, CaML,...

18. Subtypes and F : substyping

Subtyping is not really compatible with system F despite some attempts by Cardellior or Soloviev.

But subtyping is not the IS A relation that we are looking for.

Subtyping: inclusions between complex types like $a \longrightarrow b$ are all the ones derived from inclusions on a and inclusions on b.

19. Subtypes and F : substyping vs. IS A

Does subtyping on verb types derives from subtyping of its arguments, subject, object, etc. ?

Does classifications of "food"and "eaters" provide a classification of "eating" verbs (*swallow, taste, appreciate*)?

Worse:

- language does not allow all the ontological inclusion
- it is unclear whether idiosyncratic linguistic inclusions define an order

20. Afraid of impredicativity?

The definition of the type $U = \Pi X.\ T[X]$ involves among possible values for X the type U itself. One could fear to encounter paradoxes like the Russell paradox and some extensions of F are problematic, but F is not.

21. Don't be afraid of impredicativity

Two arguments:

Coherence is a consequence of normalisation and its proof use the axiom of comprehension CA (which defines a set $\{X|P(X)\}$ from a formula P) for any formula P (TT is simpler: normalisation only makes use of CA for $P \in \Pi^1_1$).

Denotational semantics It is possible to construct a category where objects are types (sets of (normal) terms of this type), morphisms from A to B corresponds to (normal) terms of type $A \rightarrow B$ (or terms of the object B^A), and the functor mapping X to $T[X]$ is isomorphic to an object that interprets $\Pi X.\ T[X]$

22. F vs. TT alternatives?

Formal complexity

- Algorithmic complexity is not an issue (syntax performs parsing, semantics β-reduces simple terms)
- Martin-Löf TT (used by Z. Luo) many rules, many variants
- Coherence: full CA vs. Π^1_1 CA (see previous slide)
- F defined by 4 rules and 2 reduction patterns

23. Dependent types and records

Records are in both system (products with named projections)

Dependent types: most TT have them, and not F but they can be added to F if one wishes to (constructions = F_{ω}+dependent types) A question: why do we need dependent types?

24. Some reflections

- System F is a very simple — but powerful — extension to the simply typed lambda calculus. This speaks in favour of our approach, provided we can account for the same data as Asher (2011).
- Do we need subtyping? A flat ontology seems rather unsatisfactory. However, coherently extending system F with subtyping is not straightforward.
- System F plus subtyping plus restricted quantification over subtypes of a type?

Part III
Generics, quantifiers

How do we express formulae with generics?
In such a way it can be derived from syntactic structures?

25. Quantifiers in syntax

Syntax, natural or logical, is preferably finitely generated (otherwise, is it syntax?)

In usual Montague semantics, with a single individual type, first order quantification has type $(\mathbf{e} \to \mathbf{t}) \to \mathbf{t}$ as soon as we have a much richer type system, we would need a quantifier per type.

In F we have a **single** constant $\forall$ of type $\Pi\alpha.\ (\alpha \to \mathbf{t}) \to \mathbf{t}$

It can be applied (specialised) to any type T to obtain the quantifier over the type T:
$\forall\{human\}(\lambda x^{human}.mortal^{human\to\mathbf{t}}(x))$.

26. The syntax of most of generic elements

Similarly we introduce

a constant $\measuredangle$ of type $\Pi\alpha.\ \alpha$

It maps each property to its specimen.

When applied to a type T, this constant $\measuredangle$ yields the element $\measuredangle\{T\}$ of type T which is assumed to be the specimen of T ($\measuredangle\{T\}$ is the F term for what one would write $\measuredangle x.T(x)$ if T were a property).

It should be observed that, as opposed to standard work, we do not say that the generalised quantifier is a property of two predicate: indeed we are in a typed version, and the first predicate of the usual setting is the type.

27. The general organisation of the lexicon, reminder

For each word in the lexicon:

- Argumental structure a second order λ-term of F
- Meaning transfers, type transformation when there is a type mismatch to encode the linguistic ontology $A \subset B$: terms of type $A \to B$.

28. Being tall (as a child) and not tall (as a human being)

We have some term and functions, with standard types: **Carlotta** $Carlotta : 2yoGirl$ (constant) a class of child (these classes are vague)

$h : 2yoGirl \rightarrow human$ (**optional λ-term**) these classes are included in the *human* class.

We can thereafter say that she is tall if she is taller that the average element in her class, an interval, and the class can be modified. But such the important point is that we can state such things and that the participate without any problem to the compositional process.

29. Being tall (as a child) and not tall (as a human being): computation

Here are the terms for:

$height : \Pi\alpha.\ (\alpha \rightarrow \mathit{float} \rightarrow \mathbf{t})$

$<: \mathit{float} \rightarrow \mathit{float} \rightarrow \mathbf{t}$

The term for *tall* below says that it is higher than any oh the heightS of the specimen. That's a possible view, to turn functions into relations for such an element.

tall $\Lambda\alpha.\lambda x^{\alpha}\forall\{\mathit{float}\}\lambda h_s^{\mathit{float}}\forall\{\mathit{float}\}\lambda h^{\mathit{float}}$
$height\{\alpha\}(\measuredangle\{\alpha\}, h_s) \wedge height\{\alpha\}(x, h) \Rightarrow h_s < h$
type of tall: $\Pi\alpha.\alpha \rightarrow \mathbf{t}$

30. A word on truth and reference

As announced we won't say anything.... but let's see why.

What properties are true of $\measuredangle\{A\}$?
the properties that most of A elements have?
most of them (without knowing which ones)?

What scalar values has it?
I would prefer that this elements has an interval of values encompassing the standard ones. .

31. A very good question

A reviewer pointed to my attention that if there are 80 persons which are 2meter tall and 20 that are 1.60 meter tall, then if the specimen has the properties that most elements have, he should be tall.

An easy answer: why should a two sorted population be considered as one type? Types should be cognitively natural.

32. A possible answer

We have the impression that its height, if any should be not a value but an interval, dropping that it is a function, encompassing the average heights.

Assume that in a species male are much taller than females, what would be the height of the "most of" generic? I would say all values from a bit less than the average female height to a bit more than the average male height.

33. Two directions to search the meaning of a specimen

But as it is well known such issues are very complicated: vagueness, comparison classes (see e.g. the book by Paul Egré).

My present idea: from a measure (in the mathematical sense) on individuals, can we define a measure on the sets of predicates defined on this set? Yes but it is quite complicated.

Other idea: give up the reference and truth and see which arguments lead to a statement with a generic, which one refute it.

34. Conclusion: type theory from syntax to DRS

Syntax (multimodal categorial grammars, automatically acquired form corpus) and semantics in λ-DRT (handwritten semantic lexicon, hence small semantics lexicon).

Using system F for varying types and licit type-flexibility including a treatment of phenomena of semantics, lexical pragmatics.

Implementation of coercion in a prototype by Emeric Kien and of quantification and "most of' generics by Samira Kherfellah.

In the mean time, we can think of the the interpretation, model theoretic or, rather, interactive, of our logical forms in particular for **specimens**.