Individuals, equivalences and quotients in type theoretical semantics.

Christian Retoré (Univ. Montpellier & LIRMM/Texte)

Léo Zaradzki (Univ. Paris Diderot & CRI & LLF)

Logic Colloquium Udine Luglio 23-28

A Introduction

A.1. What we are to speak about

Computational formalisation of the construction of meaning as logical formulae.

Fully automated in Richard Grail syntatic/semantic parser (MMCG + λ-DRT)

Formalisation: admittedly square and simplistic, but it makes things precise.

Give hints to analyse other phenomena.

Insertion of lexical semantics into compositional/formal semantics.

Sentences $\longrightarrow$ logical formulas explaining their meaning

Objects, rules: **finite** description

Semantics: **computable** map from sentences to meanings. (cognition)

A.2. General framework for compositional semantics encompassing some lexical features

Selectional restriction meaning transfers, coercions

(1) # A chair barked.

(2) Liverpool is a big place.

(3) Liverpool won the cup.

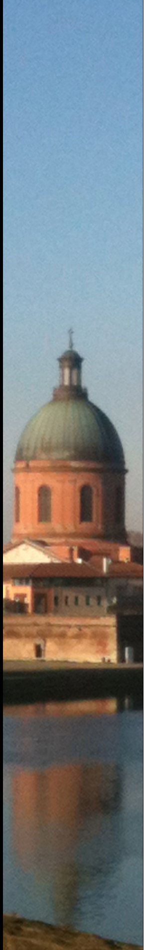
(4) Liverpool voted against having a mayor.

Felicitous and infelicitous copredications

(5) Liverpool is a big place and voted against having a mayor.

(6) # Liverpool won the cup and voted against having a mayor.

This lead us to a **rich type system.**



B Reminder on Montague semantics

B.1. A semantic lexicon

word	***semantic type*** u^* ***semantics : λ-term of type*** u^* x_v *the variable or constant* x *is of type* v
some	$(e \to t) \to ((e \to t) \to t)$ $\lambda P_{e \to t}\ \lambda Q_{e \to t}\ (\exists_{(e \to t) \to t}\ (\lambda x_e (\wedge_{t \to (t \to t)} (P\ x)(Q\ x))))$
statements	$e \to t$ $\lambda x_e (\texttt{statement}_{e \to t}\ x)$
speak_about	$e \to (e \to t)$ $\lambda y_e\ \lambda x_e\ ((\texttt{speak\_about}_{e \to (e \to t)}\ x) y)$
themselves	$(e \to (e \to t)) \to (e \to t)$ $\lambda P_{e \to (e \to t)}\ \lambda x_e\ ((P\ x) x)$

B.2. Semantic analysis

If the syntactic analysis yields:

((some statements) (themsleves speak_about)) of type t

Then one gets:

$$(\exists_{(e\to t)\to t}\ (\lambda x_e (\wedge(\mathtt{statement}_{e\to t}\ x)((\mathtt{speak\_about}_{e\to(e\to t)}\ x)x))))$$

that is to say:

$$\exists x : e\ (\mathtt{statement}(x)\ \wedge\ \mathtt{speak\_about}(x, x))$$

This is a (simplistic) semantic representation of the analysed sentence.

What about: *The chair barked* ? Needs for a richer type system.

C The Montagovian Generative Lexicon (with system F)

C.1. Types and terms

1. Constants types $\mathbf{e}_i$ and $\mathbf{t}$, as well as any type variable $\alpha, \beta, ...$ in P, are types.
2. Whenever T is a type and α a type variable which may but need not occur in T, $\Lambda\alpha.\ T$ is a type.
3. Whenever T_1 and T_2 are types, $T_1 \to T_2$ is also a type.

1. A variable of type T i.e. $x : T$ or x^T is a *term*.
   Countably many variables of each type.
2. $(f\ \ t)$ is a term of type U whenever $t : T$ and $f : T \to U$.
3. $\lambda x^T.\ t$ is a term of type $T \to U$ whenever $x : T$, and $t : U$.
4. $t\{U\}$ is a term of type $T[U/\alpha]$ whenever $t : \Lambda\alpha.\ T$, and U is a type.
5. $\Lambda\alpha.t$ is a term of type $\Lambda\alpha.T$ whenever α is a type variable, and $t : T$ without any free occurrence of the type variable α.

C.2. Using system F

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

System F with many base types $\mathbf{e}_i$ (many sorts of entities)

$\mathbf{t}$ truth values

types variables roman upper case, greek lower case

usual terms that we saw, with constants (free variables that cannot be abstracted)

Every normal terms of type $\mathbf{t}$ with free variables being logical individual and predicate constants (of a the corresponding multi sorted logic L) corresponds to a formula of L.

C.3. Co-predication

Given types α, β and γ
three predicates $P^{\alpha\rightarrow\mathbf{t}}$, $Q^{\beta\rightarrow\mathbf{t}}$, $R^{\gamma\rightarrow\mathbf{t}}$,
over entities of respective kinds α, β and γ
for any ξ with three morphisms from ξ to α, to β, and to γ

we can coordinate the properties P, Q, R of (the three images of) an entity of type ξ:

AND2= $\Lambda\alpha\Lambda\beta\Lambda\gamma$

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

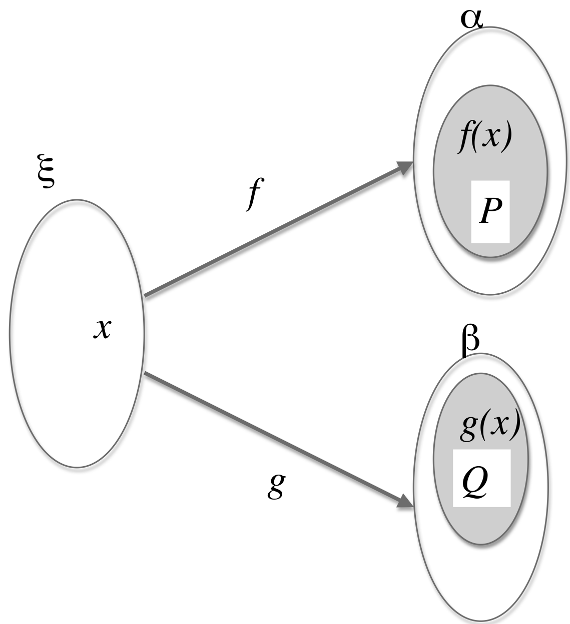


Figure 1: Polymorphic conjunction: $P(f(x))\&Q(g(x))$ with $x:\xi$, $f:\xi\to\alpha$, $g:\xi\to\beta$.

C.4. Principles of our lexicon

- Remain within the realm of Montagovian compositional semantics (for compositionality)
- Allow both predicate and argument to contribute lexical information to the compound.
- Integrate within existing discourse models (λ-DRT).

We advocate a system based on *optional modifiers*.

C.5. The Terms: main / standard term

Every lexeme is associated to an *n*-uple such as:

$$\left(\mathsf{Paris}^T, \frac{\lambda x^T.\ x^T}{\varnothing}, \frac{\lambda x^T.(f_L^{T\to L}\ x)}{\varnothing}, \frac{\lambda x^T.(f_P^{T\to P}\ x)}{\varnothing}, \frac{\lambda x^T.(f_G^{T\to G}\ x)}{rigid}\right)$$

Rigid means that when such a coercion is used, no other can be used (including the identity).

C.6. Facets (dot-objects): incorrect copredication

Incorrect co-predication. The rigid constraint blocks the copredication e.g. $f_g^{Fs \to Fd}$ cannot be **rigidly** used in

(??) *The tuna we had yesterday was lightning fast and delicious.*

C.7. Facets, correct co-predication. Town example 1/3

T town L location P people

København

k^T $f_l^{T \to L}$ $f_p^{T \to P}$

København is both a seaport and a capital.

C.8. Facets, correct co-predication. Town example 2/3

Conjunction of $cap^{T\to t}$ and $port^{L\to t}$, on k^T

If $T = P = L = e$, (as in Montague)

$(\lambda x^e((\mathrm{and}^{t\to(t\to t)}\ (\mathrm{cap}\ x))\ (\mathrm{port}\ x)))\ k.$

Conjunction between two predicates... use **AND2**

AND2= $\Lambda\alpha\Lambda\beta\Lambda\gamma$

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

f, g and h convert x to **different** types (flexible).

C.9. Facets, correct co-predication. Town example 3/3

AND2 applied to T and L and to $cap^{T\to t}$ and $\mathrm{port}^{L\to t}$ yields:

$$\Lambda\xi\lambda x^{\xi}\lambda f^{\xi\to\alpha}\lambda g^{\xi\to\beta}\lambda h^{\xi\to\gamma}.(\text{and } (cap^{T\to t}\ (f_t\ x)))(\mathrm{port}^{L\to t}\ (f_l\ x)))$$

We now wish to apply this to the type T and to the transformations provided by the lexicon. No type clash with $cap^{T\to t}$, hence $id^{T\to T}$ works. For L we use the transformations f_p and f_l.

$$(\mathrm{and}^{t\to(t\to t)}(\mathrm{cap}(\mathrm{id}\ k^{T})^{T})^{t})^{t}(\mathrm{port}\ (f_l\ k^{T})^{L})^{t})^{t}$$

If we would have conjoined a property of the place with a property of the people, instead of id we would have the map $f_l^{T\to P}$ from town T to people P from the lexicon.

(7) Kobenhavn is a capital and defeated Dortmund.

If we consider at the same time the town and the football team, the copredication is impossible because the transformation of a town into a football club $f_l^{T\to F}$ is incompatible with any other transformation even with the identity.

D The ”book” case and equivalence classes

D.1. Individuation of ”books” and multifacet object

Assume ”to read” only has the meaning of understanding, mastering (and not to decrypt signs).

(8) I carried all the books that were on the shelf to the attic because I already read them all.

Five books, including two copies of *Dubliners*.

Carried: 5

Read: 4

We do not consider the case where one books contain several books, as the Bible, which contains e.g. the book of Job.

D.2. A proper treatment in MGL

Two coercions are associated with ”book”

- f from ”book” to ϕ physical objects.
- i from ”book” to I informational objects.

”Carried” selects physical books of type ϕ, ie the $f(b)$’s.

”Read” select informational contents of books of type ϕ i.e. the $i(b)$.

So counting should apply to to the selected aspect of books (their images via coercions).

A remark for linguists: this work with E-type pronoun interpretation of ”them”, the repeated semantic term for ”them” is the one *before* any coercion is applied.

D.3. A conceptual critic

The informational content of a book may be viewed, not as a facet of the book, as a feature ”included” in the book, but as an equivalence class of books.
First or higher order predicate calculus does not include something particular to deal with quotient classes nor equivalence relations, but, given tow books b and b' one may define:

$$b \sim^{read} b' : \forall x.\ read(x, b) \leftrightarrow read(x, b')$$

This definition of $\sim^{read}$ is questionable:

1. clearly ”read” should be understood as ”understand” not as to ”decrypt signs” e.g. a page is damaged.
2. it may even be wider and vaguer than 1. because inessential differences should be left out (e.g. 1 missing page out of 500).
3. How do we use the definition? (no deductive system).

D.4. An unreachable ideal

As seen above we need to distinguish among the possible senses of ”read”, the sense ”understand” so we assume we have two lexical entries for ”read” (related in the MGL lexicon via a coercion), $\overline{read}$ (understand) and *read* (root meaning).

Ideally, one would like to define both the equivalence relation and the equivalence class $\overline{b}$ — without assuming a type/sort for texts, but defining it from ”read/understand”.

$\overline{b}$: the class of books with the same content as b, i.e. the books that are similar as far as reading is concerned.

It is impossible to define both $\overline{b}$ and $\overline{read}$ simultaneoulsy. Nevertheless each of the two may easily be defined from the other one.

An economical way to define both $\overline{b}$ and $\overline{read}$ is to assume the existence of an equivalence relation R over books such that for any two books b, b', bRb' iff $\forall x \overline{read}(x, \bar{b}) \leftrightarrow \overline{read}(x, \bar{b}')$.

D.5. Limitation of MGL

MGL (as Montague semantics) is predicate calculus (first or higher) order logic. It expresses formulas, compute them following syntax, but does not include a deductive system (nor interpretations).

There is nothing about quotients — which require canonical elements to be computable.

If we added deduction rules to MGL because of higher order, there would be a difference between true in all models and derivable — unless we use Henkin models that are not so natural.

D.6. Perspectives

Type theoretical semantics based on Martin-Lof type theory both deal with formulas and proofs, so it should be a better solution.

Some variants includes rules for dealing wuth quotients provided the classes have *canonical elements*.

Observe that for book, we have canonical objects of a different type representing the contents of books, e.g. the databases.