



Recent Advances in the Logical Representation of Lexical Semantics

Bruno Mery & Christian Retoré

Université de Bordeaux & IRIT, Toulouse

NLCS Worksop

Tulane University, New Orleans, LA

June 28th 2013



Topic of this talk

Pursuing the enhancement of analysers for syntax and semantics with compositional models of the lexicon.

- Compositional Semantics in the NLP Analysis
- Our Framework for Lexical Semantics
- The Types and Sorts
- Recent Advances



A – Formal Semantics and Pragmatics in NL Analysers



A.1. Montague Grammar

Syntax-driven composition of semantics

- Syntax (surface phenogrammatcs), Semantics (logical meaning)
- Deep structure (tectogrammatcs) representations
- Analysis of Syntax → templates for Composition of Semantics

Two logics for the analysis

- Meaning representation: logical formulae (e.g., first-order predicate logic)
- Meaning assembly / “Glue”: **proofs** of the formulae (intuitionist propositional logic, λ -calculus)



Types in Montague

- **e**: single **sort** of entities
- **t**: truth-valued formulae
- Possibly indices, events. . .

Syntax and Semantics Analysers

- **Boxer** (Johan Bos), **Grail** (Richard Moot). . .
- Statistically acquired lexicon and grammar
- Based on **categorial grammars**
- Relying on Montague Grammar

(Syntactic type)*	=	Semantic type
S^*	=	t a sentence is a proposition
np^*	=	e a noun phrase is an entity
n^*	=	$e \rightarrow t$ a noun is a subset of the set of entities
$(A \setminus B)^* = (B / A)^*$	=	$A \rightarrow B$ extends easily to all syntactic categories of a Categorial Grammar e.g. a Lambek CG



A.2. Applying the Montagovian Analysis

Procedure

1. Put corresponding terms from the lexicon in place of words
2. Reduce the λ -term to its normal form: a truth-valued formula

Illustration

<i>some</i>	$(e \rightarrow t) \rightarrow ((e \rightarrow t) \rightarrow t)$ $\lambda P^{e \rightarrow t} \lambda Q^{e \rightarrow t} (\exists^{(e \rightarrow t) \rightarrow t} (\lambda x^e (\wedge^{t \rightarrow (t \rightarrow t)} (P x)(Q x))))$
<i>club</i>	$e \rightarrow t$ $\lambda x^e (\text{club}^{e \rightarrow t} x)$
<i>defeated</i>	$e \rightarrow (e \rightarrow t)$ $\lambda y^e \lambda x^e ((\text{defeated}^{e \rightarrow (e \rightarrow t)} x)y)$
<i>Leeds</i>	e Leeds



Syntactical structure: (some (club)) (defeated Leeds)

$$\begin{aligned}
 & \left(\left(\lambda P^{e \rightarrow t} \lambda Q^{e \rightarrow t} \left(\exists^{(e \rightarrow t) \rightarrow t} \left(\lambda x^e \left(\wedge (P x) (Q x) \right) \right) \right) \right) \left(\lambda x^e \left(\text{club}^{e \rightarrow t} x \right) \right) \right) \\
 & \quad \left(\left(\lambda y^e \lambda x^e \left(\left(\text{defeated}^{e \rightarrow (e \rightarrow t)} x \right) y \right) \right) \text{Leeds}^e \right) \\
 & \quad \quad \quad \downarrow \beta \\
 & \left(\lambda Q^{e \rightarrow t} \left(\exists^{(e \rightarrow t) \rightarrow t} \left(\lambda x^e \left(\wedge^{t \rightarrow (t \rightarrow t)} \left(\text{club}^{e \rightarrow t} x \right) \left(Q x \right) \right) \right) \right) \right) \\
 & \quad \left(\lambda x^e \left(\left(\text{defeated}^{e \rightarrow (e \rightarrow t)} x \right) \text{Leeds}^e \right) \right) \\
 & \quad \quad \quad \downarrow \beta \\
 & \left(\exists^{(e \rightarrow t) \rightarrow t} \left(\lambda x^e \left(\wedge \left(\text{club}^{e \rightarrow t} x \right) \left(\left(\text{defeated}^{e \rightarrow (e \rightarrow t)} x \right) \text{Leeds}^e \right) \right) \right) \right)
 \end{aligned}$$

Usually human beings prefer to write it like this:

$$\exists x : e \left(\text{club}(x) \wedge \text{defeated}(x, \text{Leeds}) \right)$$



A.3. Restriction of Selection and Polysemy

Selection

- Predicates (syntactically) select arguments
- The **lexical field** of those arguments is restricted
- Other arguments can be **forced** to behave as expected

Differences in acceptability

- The dog barked.
- The chair barked.
- The drill sergeant barked.
- The hawker barked.



Contrastive ambiguity

- Different lexemes, homophonic / homographic
- Bank, Bar, Pen. . .

Relational polysemy

- A single word, different uses and related meanings
- The bank killed my account.
- The school is on strike.



A.4. Acceptability and Felicity: Semantics, Pragmatics or Both ?

Montague: everything is acceptable

- All syntactically valid items have the same semantic “meaning”
- We have to rely on pragmatics or interpretation

Strong restriction from lexical semantics

- e is replaced by many sorts
- Barking dogs are licensed, everything else is blocked
- No language works that way

Creative uses and semantic licenses

- **Fast** runners, cars, computers. . . and phones
- **A delicious game** (Cooper)
- **Expertly built** (Adams)



Pragmatic licenses

- Variations from the lexicon in specific contexts external to the utterances
- I deceased again last evening
- The analyser cannot be expected to get a correct meaning without the external context

Our position on the integration of “Pragmatics”

- Creative use should be included
- Anything that can be comprehended in a self-contained text should be included
- We should not try to account for unknown contextual information
- World knowledge, background, social and universal contexts can be integrated
- Pragmatics should not be a pretext to give up on critical variations of meaning



B – Our Generative Framework



B.1. Foundations

The Generative Lexicon

- Pustejovsky, 1995 (and precursors)
- Discussed and refined by Asher, Cooper, Luo, Nunberg, Partee. . .
- Idea: the lexicon provides enough data to **generate** word meaning in context

Phenomena covered

- Contrastive ambiguity
- Hyponymy / Hyperonymy
- Qualia (properties, agent, purpose, component)
- Grinding / Packing
- Facettes
- Co-predications



Type system

- TY_n
- Different lexical behaviours imply different base types
- How detailed ? (later)

Framework sketch

- Lexical entries are typed with **many** sorts
- Each word has a single **main** λ -term
- Each word can have any number of **optional** λ -terms
- Those terms are **modifiers**
- Normal application is the same
- Modifiers are used when types **clash**
- Types **guide** the selection of modifiers



B.2. Second-Order Calculus

Many-sorted logic $\wedge TY_n$, λ -calculus with second-order types

System-F The simplest implementation of second-order types

Simple examples

Normal application

small stone

$$\overbrace{(\lambda x^\varphi. (\text{small}^{\varphi \rightarrow \varphi} x))}^{\text{small}} \overbrace{\tau^\varphi}^{\text{stone}}$$

$$(\text{small } \tau)^\varphi$$

Qualia exploitation

wondering, loving smile

$$\overbrace{(\lambda x^P. (\text{and}^{t \rightarrow (t \rightarrow t)} (\text{wondering}^{P \rightarrow t} x) (\text{loving}^{P \rightarrow t} x)))}^{\text{wondering, loving}} \overbrace{\tau^S}^{\text{smile}}$$

$$(\lambda x^P. (\text{and}^{t \rightarrow (t \rightarrow t)} (\text{wondering}^{P \rightarrow t} x) (\text{loving}^{P \rightarrow t} x)))) (f_a^{S \rightarrow P} \tau^S)$$

$$(\text{and } (\text{loving } (f_a \tau)) (\text{loving } (f_a \tau)))$$



B.3. Conjunction for co-predication

! Polymorphic conjunction:

Given predicates $P^{\alpha \rightarrow \mathbf{t}}$, $Q^{\beta \rightarrow \mathbf{t}}$ over entities of respective types α , β ,
given any type ξ with two morphisms from ξ to α , to β
we can coordinate the properties P , Q of (the two images of) an entity of type ξ :

The polymorphic conjunction $\&^\Pi$ is defined as the term

$$\begin{aligned} \&^\Pi = \Lambda \alpha \Lambda \beta \lambda P^{\alpha \rightarrow \mathbf{t}} \lambda Q^{\beta \rightarrow \mathbf{t}} \\ \Lambda \xi \lambda x^\xi \lambda f^{\xi \rightarrow \alpha} \lambda g^{\xi \rightarrow \beta}. \\ (\text{and}^{\mathbf{t} \rightarrow \mathbf{t} \rightarrow \mathbf{t}} (P (f x))(Q (g x))) \end{aligned}$$

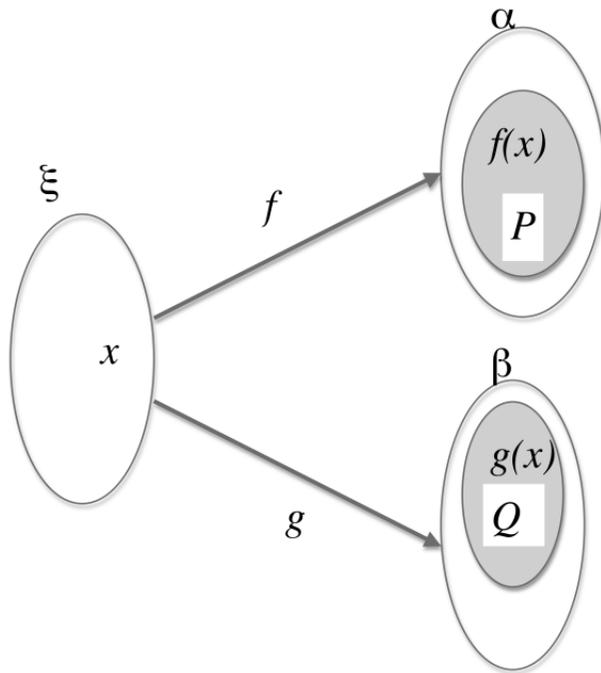


Figure 1: Polymorphic conjunction: $P(f(x)) \& Q(g(x))$
with $x : \xi$, $f : \xi \rightarrow \alpha$, $g : \xi \rightarrow \beta$.



B.4. Constraints

Possible/Hazardous co-predicative constructions

- Important point: this is not (only) about toy examples.
- *The salmon was fast and delicious.
- The salmon was lightning fast. It is delicious.
- Liverpool is a large city and voted labour.
- *Liverpool is a large city and won the cup.

Constraints on flexibility

- Are lexically fixed on modifiers, computed for terms
- Flexible: anything goes.
- Rigid: nothing else can go (even the original typing).

Complex examples



Lexicon

word	principal λ -term	optional λ -terms	rigid/flexible
<i>Liverpool</i>	<i>liverpool</i> ^T	$Id_T : T \rightarrow T$ (F) $t_1 : T \rightarrow F$ (R) $t_2 : T \rightarrow P$ (F) $t_3 : T \rightarrow Pl$ (F)	
<i>is_a_big_place</i>	<i>big_place</i> : $Pl \rightarrow \mathbf{t}$		
<i>voted</i>	<i>voted</i> : $P \rightarrow \mathbf{t}$		
<i>won</i>	<i>won</i> : $F \rightarrow \mathbf{t}$		

where the base types are defined as follows:

- T town
- F football club
- P people
- Pl place



Liverpool is a big place and voted

Polymorphic AND yields:

$$(\&^{\Pi}(big_place)^{Pl \rightarrow t}(voted)^{P \rightarrow t})$$

Forces $\alpha := Pl$ and $\beta := P$, the properly typed term is

$$\&^{\Pi}\{Pl\}\{P\}(big_place)^{Pl \rightarrow t}(voted)^{P \rightarrow t}$$

It reduces to:

$$\Lambda\xi\lambda x^{\xi}\lambda f^{\xi \rightarrow \alpha}\lambda g^{\xi \rightarrow \beta}(\text{and}^{t \rightarrow t} \rightarrow t (big_place (f x))(voted (g x)))$$

Syntactical relaxation of semi-flexible constraints

- The salmon was lightning fast. It is delicious.
- **Semi-flexible**: acts as **Rigid**, but is reset by reference.



B.5. Quantification

Quantification in Montague Grammar

- Common nouns are **predicates**, applying articles yields individuals
- $cat : e \rightarrow t$, $a : (e \rightarrow t) \rightarrow e$, $a\ cat : e$
- Denotations of **a** v. **the**, **some**, **most**...

Issues with polymorphic quantification

- With **stone**: $\varphi \rightarrow t$, **man**: $A \rightarrow t$, **cat**: $Animal \rightarrow t \dots$
- ... what is the type of **a** ?



Nouns: base types or predicates ?

- Many frameworks have failed because they assume nouns are predicates. Treating them as base types makes things a great deal easier (Luo).
- We want to retain the predicate **the property of being of type** x .
- Why not use both, with an appropriate construct, \hat{x} ?

The use of Hilbert's operators for determiners

- Russel: $\iota_x P(x)$ (unique x satisfying $P(x)$)
- Hilbert: ε_x (existing) and τ_x (universal)
- von Heusinger: use of ε for determiners
- ε_x^1 : most salient x

ε and polymorphism

- Nouns as predicates: ε is of type $\Lambda\alpha.(\alpha \rightarrow \mathbf{t}) \rightarrow \alpha$
- Nouns as base types: ε is of type $\Lambda\alpha.\alpha$



C – The Type System



C.1. Different approaches

The singular sort (and pragmatics) The lexicon is empty.
Meaning is to be constructed at a later stage

A few different sorts Noun classes, or dictionary RSs:
physical/abstract, animate/inanimate, artificial/natural. . .

Many different sorts
Identifying which requires a thorough methodology.

All nouns are sorts
A complete hierarchy with subtyping relations (Luo). Can
be derived from lexical ontologies.

All formulae are sorts
Every formula with a single free variable, $F(x)$. In single-
sorted logic, or the definition becomes circular.

Ad-hoc sorts
Generate what is needed for every application.



C.2. The Existing Implementation

The syntax and semantics analyser GRAIL

- Richard Moot's established syntax and semantics parser
- Wide-coverage, based on categorial grammars
- Acquires lexicon and rules statistically
- PROLOG-base, active development
- Good results on spoken French

Logical programming and additional types

- No issues when changing the type system
- Second-order modifiers as a module
- Actual implementation in λ -DRT rather than λ -calculus



Partial, functioning implementations

- A proof of concept in 2010 for modifiers
- More recently: application to a specific problem
- Complete, working implementation for a domain
- Travel stories in the Pyrenean region, late *XIXth*
- Sorts distinguish between places, times, itineraries. . .

What is missing

- Given a type system, Grail can infer the lexicon
- We need a comprehensive type system



C.3. The Classifier Approach

Features of the classifier systems

- Common to several language families
- Asian languages, Sign languages. . .
- Used for counting and measuring
- Apply to most entities in the lexicon

Granularity

- Many classifiers, but a lot less than the lexicon
- Semi-organized: generic, common, specialized
- Variations of use and lexical licenses
- Naturally occurring

Implicit sorts in English and French In progress



D – Other Discussions



D.1. Linear Constraints

Current constraints

- Flexible (compatible with everything)
- Rigid (not compatible with anything)
- Plus syntax-based relaxation

Possible configurations

- Set A modifiers compatible with each others, but not with set B
- Using a modifier f_1 imply we may not use a modifier f_2 , except if another g is used first
- ... and even more complicated constraints
- **This is not a priority**, as we have no examples



Linear Formulation

- Idea: associate to lexemes a **linear formula** describing the available modifiers (rather than a list)
- If possible sets are $\{k\}$, $\{f, g\}$, $\{g, h\}$, $\{h, f\}$, excluding any other configurations. . .
- That formula is: $!k \& (!f \otimes !g) \& (!g \otimes !h) \& (!h \otimes !f)$



D.2. Felicity

- Infelicitous statements can receive a meaning from context
- In most applications, meaningfulness should be assumed
- (Exceptions include auto-completion and dictation !)
- When a type clash is not solvable:
 1. generate an appropriate modifier
 2. try to infer its properties



D.3. Strong Idiosyncrasy

Linguistic constructs are not all cognitively motivated
idioms and specific constructs are illustrations of this.

Differences of language

I got punctured

Differences of dialect

Half a strawberry

Differences of jargon

Straighten #16



D.4. Contexts and Variations of Lexica

Variable Lexica and the need to be adaptable

Idioms from language, dialect, and jargon As seen

Literary settings

- Some narratives assume background knowledge from a specific period / location
- Tales, Fables, Fantasy, SF... Routinely use an expanded lexicon
- Sometimes types are overloaded (Animals as Agents)



Example with implicit introduction (Lewis Carroll)

“Beware the Jabberwock, my son!
The jaws that bite, the claws that catch!
Beware the Jubjub bird, and shun
The frumious Bandersnatch!”

Example with explicit introduction (George Martin)

Their driver awaited them beside his *hathay*.
In Westeros, it might have been called an oxcart,
though it [...] lacked an ox.
The *hathay* was pulled by a dwarf elephant[...]

Example with in-character introduction (J. K. Rowling)

‘I’d like ter see a great Muggle like you stop him,’
he said.
‘A what?’ said Harry, interested.
‘A Muggle,’ said Hagrid.
‘It’s what we call non-magic folk like them.’



E – Per Aspera



E.1. Where we stand

A motivated type-theoretical framework in ΛTY_n

A functioning implementation in Grail

Theoretical and practical applications Plurals, virtual travellers,
travel stories

Continuing work On the type system, quantification. . .

Where to go from here

- More data
- More discussions
- Implementation, validation, evaluation



E.2. Some References

Work presented in this talk:

The Montagovian generative lexicon ΛT_{y_n} : an integrated type-theoretical framework for compositional semantics and lexical pragmatics <http://hal.archives-ouvertes.fr/hal-00779214>

(with Michele Abrusci) Some proof theoretical remarks on quantification in ordinary language <http://hal.archives-ouvertes.fr/hal-00779223>

A lexicon for compositional semantics and lexical pragmatics: <http://hal.inria.fr/inria-00408308> article in the Journal of Logic, Language and Information, 2010 (Ch. Bassac, B. Mery, Ch. Retoré)

Fictive motion, virtual traveller <http://hal.inria.fr/hal-00650635> in French with plain Montagovian λ -terms (TALN 2011) or <http://hal.archives-ouvertes.fr/hal-00607691> in English with λ -DRT (CID 2011) (R. Moot, L. Prévot, Ch. Retoré)

Quantification: <http://hal.archives-ouvertes.fr/hal-00677312> "most" in this setting (article in RLV) (Ch. Retoré)

Plurals: <http://hal.inria.fr/hal-00650644> a talk at the Coconat workshop (R. Moot, Ch. Retoré)



Related work:

Nicholas Asher [Lexical meaning in context: a web of words](#). Cambridge University Press 2011.

Zhaohui Luo Contextual Analysis of Word Meanings in Type-Theoretical Semantics in [Logical Aspects of Computational Linguistics](#) 2012, Springer LNCS 6736 edited by S. Pogodalla and J-Ph. Prost.