White Fang is not a dog. What about Dumbo?

Léo Zaradzki

Logic Colloquium 2018, Udine

Presentation outline

1. Elements of linguistics

2. Existing approaches

3. Proposal and Perspectives

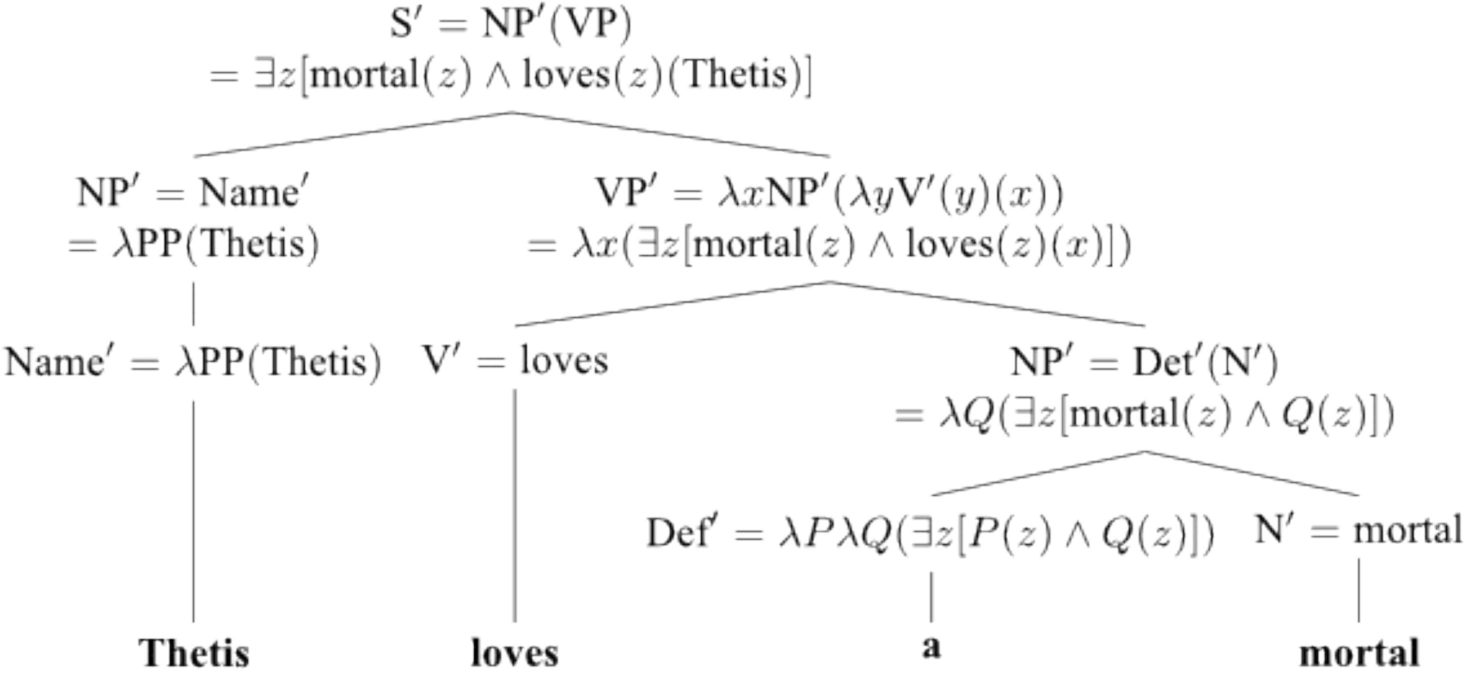
Compositional Semantics

Semantics:

- Takes the syntactic structure of a sentence as an input
- Applies formal rules to derive the meaning of the sentence, in a compositional way
- Returns a semantic representation of this meaning

Classical view: Knowing the meaning of a sentence is knowing its truth-conditions.

Example: “Thetis loves a mortal”



Montague Semantics

- Based on simple type theory: two basic types (e and t), and one constructor ($\rightarrow$)
- Common nouns and intransitive verbs are both represented as predicates (of type $e \rightarrow t$).

A few problems (among others):

- “The television ate a cake” is a well-formed sentence.
- Donkey anaphoras like “If John owns a donkey, he beats it” can’t be represented in a compositional way.
- Model can’t distinguish between sense and reference.

Martin-Löf Type Theory

- It's an alternative to set theory. Basic objects are types.
- The proposition "$a = b$" is well-formed only if a and b are of the same type (in fact, there is an equality relation for each single type).
- There is a (1-1) correspondence between types and logical propositions. In particular, for objects a, b of some type A, the proposition "$a = b$" is itself a type.
- An inhabitant of a type is a proof of the corresponding proposition.

Semantics based on Rich Type Theories

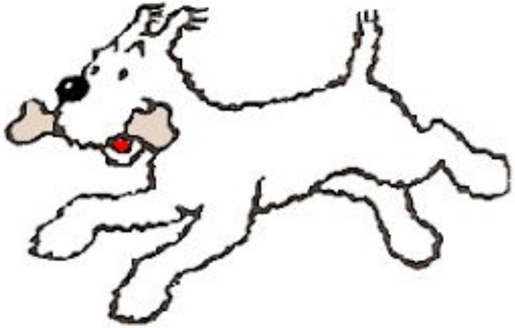
- More basic types (like in many-sorted logics)
- More constructors (*e.g.* Pi-types, Sigma-types, ...)

- Strong typing prevents nonsense (just like in computer science) and helps with copredication (Chatzikyriakidis, Luo).
- Dependent function types help account for donkey anaphoras (Ranta).
- Intensionality helps distinguish between sense and reference.

Question : What should we take for basic types?

Common Nouns as Types

- There is a basic type for each common noun.
- A verb like *bark* is a function **Dog** → **Prop**.



- "Milou is a dog" is represented as the judgement m : **Dog**.
- For each type A, there is a predicate p_A : A → **Prop** which says "I am of type A".

CN as Types and Negation

Problem: In Type Theory you can't negate judgements!

In particular, these are not well-formed sentences:

- "Milou is not a dog".
- "The chair didn't bark" (not clear whether this should be an acceptable sentence).
- "If Milou is a dog, he likes bones".

Chatzikyriakidis and Luo’s Solution

They assume the existence of an operator:

$$\mathrm{NOT} : \prod_{A:CN} \ \prod_{p:A \to Prop} \ \prod_{B:CN} \ \prod_{b:B} Prop$$

such that NOT(*A*, *p*, *B*, *b*) means “*b* does not *p*” (possibly with *B* different from *A*).

They stated a few axioms for NOT and proved it to be consistent.

See, among others : S. Chatzikyriakidis, Z. Luo and T. Xue, *Propositional Forms of Judgmental Interpretations*, 2018.

A few criticisms against this view

- Unwanted sentences like “Milou is not even” or “The number 2 is not red” become acceptable;
- Subtyping problems;
- The case of synonymy (*e.g.* chien/clébard/cabot);
- Syntactical problems:
  - Modified nouns;
  - Determiners;
- Complexity problems:
  - Space complexity;
  - Type hierarchy.

Classifiers as Types

- Classifiers form a category of words in some languages (many Asian languages, all sign languages, *etc.*)
- *Example*: Japanese “Hon” is used to count long things, including rivers, roots, but also things that take long time such as movies, tennis matches.
- Typically a few hundreds of classifiers in a language.
- Classifiers can be used to group objects that have in common the possibility of being applied to the same predicate.

The "Comparable-Likely" Operator S

- For each basic type *A*, we assume the existence of a type *S(A)* "containing" *A*.
- Intuitively, *S(A)* contains the objects that are likely to be mistaken for inhabitants of *A*.
- There are coercions $A \to S(A)$ and equalities $S(S(A)) = S(A)$.
- Now common nouns are predicates defined on the *S*-closure of their classifiers.
  Example: If there is a classifier **Canidae**, then *dog* and *canidae* are both predicates of type *S*(**Canidae**) → **Prop**.

Perspectives

- Formalise and axiomatise the operator S.
- Find out whether this is coherent with the CN-as-Types or the Classifiers-as-Types approaches.
- The genericity puzzle:
  - "The chair didn't bark".
  - "Chairs don't bark".

Thank you for your attention!

contact: leozaradzki@gmail.com

Some References:
R. Montague, *The Proper Treatment of Quantification in Ordinary English*, 1973
A. Ranta, *Type-theoretical Grammar*, 1994
N. Asher, *Lexical meaning in context - a web of words*, 2011
C. Retoré, *The Montagovian Generative Lexicon ΛTYn: a type-theoretical framework for natural language semantics*, 2013
B. Mery and C. Retoré, *Classifiers, Sorts and Types in Lexical Compositional Semantics*, 2017
S. Chatzikyriakidis, Z. Luo and T. Xue, *Propositional Forms of Judgmental Interpretations*, 2018