Logical Structures in Natural Language: Language

RAFFAELLA BERNARDI

UNIVERSITÀ DI TRENTO

E-MAIL: BERNARDI@DISI.UNITN.IT
## Contents

1  Aristotle, Stoics and Frege. ................................................................. 4  
1.1  Frege: logical vs. grammatical form .............................................. 5  
1.2  Wittgenstein and Tarski ................................................................. 6  
1.3  Frege: saturated vs. unsaturated expressions ................................... 7  
1.4  Pioneers .......................................................................................... 8  
2  Formal Semantics for NL: Main questions ........................................... 9  
2.1  Logical Approach ............................................................................. 10  
2.2  Sum up: Formal Semantics for Natural Language .............................. 11  
2.3  Example (Set Theory) ....................................................................... 12  
2.4  From sets to functions NEW ............................................................ 13  
2.5  Summing up ..................................................................................... 14  
3  Recall: Formal Semantics: What .......................................................... 15  
4  Formal Semantics: How ...................................................................... 16  
4.1  Formal Semantics: How (cont’d) ..................................................... 17  
4.2  Formal Semantics: How (Cont’d) .................................................... 18  
4.3  Compositionality ............................................................................. 19  
4.4  Ambiguity ......................................................................................... 20
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5</td>
<td>FOL: How?</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td>Building Meaning Representations</td>
<td>22</td>
</tr>
<tr>
<td>6</td>
<td>Function and lambda terms (NEW)</td>
<td>23</td>
</tr>
<tr>
<td>6.1</td>
<td>Formal Semantics: How</td>
<td>24</td>
</tr>
<tr>
<td>7</td>
<td>Done to be done</td>
<td>25</td>
</tr>
</tbody>
</table>
1. Aristotle, Stoics and Frege.

- Aristotelian were interested in the relations between the terms within premises and conclusion of a given argument (Syllogism: “All A are B”, “All B are C”, hence “All A are C”).

- Stoics focused on the conditional relation “If ... then”.

- In ’900 there is the merge of these two traditions with the introduction of quantifiers by Frege.

\[ \forall x (\text{Italian } x \rightarrow \text{Talkative } x) \]

\[ \forall x (\text{Talkative } x \rightarrow \text{Funny } x) \]

Hence, \[ \forall x (\text{Italian } x \rightarrow \text{Funny } x) \]

Furthermore, thanks to the symbols introduced by Frege, it’s possible to represent sentences with more than one quantifier.
1.1. Frege: logical vs. grammatical form

“A natural number bigger than every natural number”.

1. $\forall x \exists y \text{Bigger}(y, x)$
2. $\exists y \forall x \text{Bigger}(y, x)$

1. is true, whereas 2. is false.

The different interpretation of the sentence is given by the different scope of quantifiers. Frege distinguishes:

- Grammatical form (subject-predicate)
- Logical form (function-argument)
1.2. Wittgeinstein and Tarski

Wittgeinstein considered truth-value conditions for complex statements built by means of logical connectives, but he had not looked at truth-conditional of simple quantified sentences.

Tarski gives a precise definition for these sentences too by introducing:

- model
- domain
- interpretation function
- satisfiability
- assignment
1.3. Frege: saturated vs. unsaturated expressions

German mathematician, logician and philosopher. He wanted to develop an ideography (a formal language) to overcome natural language limitations (ambiguities).

Frege generalizes the concept of functions and applied it to linguistic expressions: e.g., Woman(x), and if we replace the variable x with a constant e.g. r, we obtain Woman(r) = true

**Saturated vs. unsaturated expressions** He distinguishes expressions in saturated (e.g., a sentence) and unsaturated (e.g., a concept).

“Caeser conquered Gaul”. “Caeser” is a complete (saturated) expression and “(·) conquered Gaul” is an unsaturated expression – it needs to be completed.

Aristotle focused on predicate-argument structure, whereas Frege introduces the distinction function vs. argument.

**First and higher order functions** Functions differ w.r.t. the nr of their arguments, moreover, they can take as argument objects or other functions. The former are called first order functions, the latter second order functions.
1.4. Pioneers

Gottlob Frege  Frege aims to avoid having to use natural language.

- Linguistics expressions can be divided into complete vs. not-complete.
- Proper name and sentences are complete (entity and truth value)
- A concept is not-complete, it’s a one-argument function
- A transitive verb is not-complete, it’s a two-argument function
- A quantifier phrase is not-complete, it’s a higher order functions.
- Logical vs. Grammatical form.

Richard Montague  Montague aims to define a model-theoretic semantics for natural language. He treats natural language as a formal language:

- Syntax-Semantics go in parallel.
- It’s possible to define an algorithm to compose the meaning representation of the sentence out of the meaning representation of its single words.
2. **Formal Semantics for NL: Main questions**

The main questions are:

1. What does a given sentence mean?
2. How is its meaning built?
3. How do we infer some piece of information out of another?

The first and last question are closely connected.

In fact, since we are ultimately interested in understanding, explaining and accounting for the entailment relation holding among sentences, we can think of *the meaning of a sentence as its truth value*, as logicians teach us.
2.1. Logical Approach

To tackle these questions we will use Logic, since using Logic helps us answering the above questions at once.

1. Logics have a precise semantics in terms of models — so if we can translate/represent a natural language sentence $S$ into a logical formula $\phi$, then we have a precise grasp on at least part of the meaning of $S$.

2. Important inference problems have been studied for the best known logics, and often good computational implementations exist. So translating into a logic gives us a handle on inference.

When we look at these problems from a computational perspective, i.e. we bring in the implementation aspect too, we move from Formal Semantics to Computational Semantics.
2.2. Sum up: Formal Semantics for Natural Language

We will exploit

- The principle of Compositionality [Frege]
- The connection between Syntax and Semantics [Montague]
- Set theory to represent the meaning of words and phrases.
- The relation between a set and its characteristic function. [NEW!]
- \( \lambda \)-Terms (and FOL) to represent functions capturing linguistic expressions. [NEW!]
2.3. Example (Set Theory)

Let our model be based on the set of entities $D_e = \{\text{lori, ale, sara, pim}\}$ which represent Lori, Ale, Sara and Pim, respectively. Assume that they all know themselves, plus Ale and Lori know each other, but they do not know Sara or Pim; Sara does know Lori but not Ale or Pim. The first three are students whereas Pim is a professor, and both Lori and Pim are tall. This is easily expressed set theoretically. Let $[[w]]$ (it’s like $I$ of Logic) indicate the interpretation of $w$:

\[
[[\text{sara}]] = \text{sara};
\]
\[
[[\text{pim}]] = \text{pim};
\]
\[
[[\text{lori}]] = \text{lori};
\]
\[
[[\text{know}]] = \{\langle \text{lori, ale} \rangle, \langle \text{ale, lori} \rangle, \langle \text{sara, lori} \rangle, \\
\langle \text{lori, lori} \rangle, \langle \text{ale, ale} \rangle, \langle \text{sara, sara} \rangle, \langle \text{pim, pim} \rangle\};
\]
\[
[[\text{student}]] = \{\text{lori, ale, sara}\};
\]
\[
[[\text{professor}]] = \{\text{pim}\};
\]
\[
[[\text{tall}]] = \{\text{lori, pim}\}.
\]

In words, e.g. the relation know is the set of pairs $\langle \alpha, \beta \rangle$ where $\alpha$ knows $\beta$; or that ‘student’ is the set of all those elements which are a student.

**Denotation vs. expression** Note, the lexical entry determine that the denotation of e.g.the English name sara is the person sara.
2.4. From sets to functions NEW

A set and its characteristic function amount to the same thing:

if \( f_X \) is a function from \( Y \) to \( \{F, T\} \), then \( X = \{y | f_X(y) = T\} \). In other words, the assertion ‘\( y \in X \)’ and ‘\( f_X(y) = T \)’ are equivalent.

\[
[\text{student}] = \{t, a, f, j\}
\]

\text{student} can be seen as a function from entities to truth values

We shift from the relational to the functional perspective.

The two notations \( (F(z))(u) \) and \( F(u, z) \) are equivalent.

Functions can be expressed by lambda terms. More in a bit!
2.5. Summing up

Summarizing, when trying to formalize natural language semantics, at least two sorts of objects are needed to start with: the set of truth values \( t \), and the one of entities \( e \).

Moreover, we spoke of more complex objects as well, namely functions. More specifically, we saw that the kind of functions we need are truth-valued functions (or boolean functions).

Hence, we need domains of entities \( (D_e) \), domains of truth values \( (D_t) \), and domains of functions, e.g. from \( e \rightarrow t \) \( (D_{e \rightarrow t}) \).

Furthermore, we have illustrated how one can move back and forwards between a set relational and a functional perspective. The former can be more handy and intuitive when reasoning about entailment relations among expressions; the latter is more useful when looking for lexicon assignments.

What does a given sentence mean?

The meaning of a sentence is its truth value. Hence, this question can be rephrased in “Which is the meaning representation of a given sentence to be evaluated as true or false?”

- **Meaning Representations**: Predicate-Argument Structures are a suitable meaning representation for natural language sentences. E.g.
  
  the meaning representation of “Vincent loves Mia” is \( \text{loves(vicent, mia)} \)

  whereas the meaning representation of “A student loves Mia” is \( \exists x. \text{student}(x) \land \text{loves}(x, \text{mia}) \).

- **Interpretation**: a sentence is taken to be a proposition and its meaning is the truth value of its meaning representations. E.g.
  
  \[ \llbracket \exists x. \text{student}(x) \land \text{left}(x) \rrbracket = 1 \text{ iff standard FOL (First Order Logic) definitions are satisfied.} \]
4. Formal Semantics: How

How is the meaning of a sentence built?

To answer this question, we can look at the example of “Vincent loves Mia”. We see that:

- “Vincent” contributes the constant \textit{vincent}
- “Mia” contributes the constant \textit{mia}
- “loves” contributes the relation symbol \textit{loves}

This observation can bring us to conclude that the \textit{words} making up a sentence contribute all the bits and pieces needed to build the sentence’s meaning representation.

In brief, \textit{meaning flows from the lexicon}. 
4.1. Formal Semantics: How (cont’d)

But,

1. Why the meaning representation of “Vincent loves Mia” is not \text{love}(\text{mia}, \text{vincent})? 

2. What does “a” contribute to in “A student loves Mia”?

As for 1., the missing ingredient is the \textit{syntactic structure}! \text{[Vincent [loves, \text{Mia}_{np}]_{vp}]}_{s}. 


Briefly, *syntactic structure guiding gluing.*
4.3. Compositionality

The question to answer is: “How can we specify in which way the bit and pieces combine?”

1. Meaning (representation) ultimately flows from the lexicon.

2. Meaning (representation) is obtained by making use of syntactic information.

3. The meaning of the whole is function of the meaning of its parts, where “parts” refer to substructures given us by the syntax.
4.4. Ambiguity

A single linguistic sentence can legitimately have different meaning representations assigned to it.

For instance, “John saw a man with the telescope”

a. John [saw [a man [with the telescope]_{pp}}_{np}\}_{vp} \exists x.\text{Man}(x) \land \text{Saw}(j,x) \land \text{Has}(x,t)

b. John [[saw [a man]_{np}}_{vp} [with the telescope]_{pp}\}_{vp} \exists x.\text{Man}(x) \land \text{Saw}(j,x) \land \text{Has}(j,t)

Different parse trees result into different meaning representations!
4.5. FOL: How?

Problems with the how:

**Constituents:** it cannot capture the meanings of constituents.

**Assembly:** it cannot account for meaning representation assembly.
5. Building Meaning Representations

To build a meaning representation we need to fulfill three tasks:

**Task 1** Specify a reasonable *syntax* for the natural language fragment of interest.

**Task 2** Specify semantic representations for the *lexical items*.

**Task 3** Specify the *translation* of constituents *compositionally*. That is, we need to specify the translation of such expressions in terms of the translation of their parts, parts here referring to the substructure given to us by the syntax.

Moreover, when interested in Computational Semantics, all three tasks need to be carried out in a way that leads to computational implementation naturally.
6. Function and lambda terms (NEW)

Recall: Function $f : X \rightarrow Y$. And $f(x) = y$ e.g. $SUM(x, 2)$ if $x = 5$, $SUM(5, 2) = 7$.

- $\lambda x. x$
- $\lambda x. (x + 2)$
- $\left(\lambda x. (x + 2)\right) 5$
- $\left(\lambda x. (x + 2)\right) 5 = 5 + 2$

student: $D_e \rightarrow D_t : \lambda x. \text{student}(x)$

Lambda calculus was introduced by Alonzo Church in the 1930s as part of an investigation into the foundations of mathematics.
6.1.  **Formal Semantics: How**

Vincent loves Mia: (S)
\[ \text{loves(} \text{vincent, mia)} \]

\[ \text{loves } \lambda y. \text{loves(y, mia)} \]

\[ \text{loves } \lambda x. \lambda y. \text{loves(y, x)} \]

\[ \text{Mia} \]

\[ \text{mia} \]

**syntactic structure guiding gluing** and the linguistic composition amounts to function application. More tomorrow.

Exercises on the syntax of lambda terms.
7. Done to be done

Today we have

- introduced the Formal Semantics approach, and motivate the need to extend FoL with lambda-terms.
- started practicing with lambda-terms.

Next time, we will

- practice with lambda calculus.
- apply lambda calculus to build FoL a meaning representation of a NL sentence.