Computational Linguistics: Semantics II

Raffaella Bernardi

KRDB, Free University of Bozen-Bolzano

Via della Mostra 4, Room: 1.06, e-mail:

bernardi@inf.unibz.it

Contents

1. Historical Remark: Montague

Montague ('74) was the first to seriously propose and defend the thesis that the relation between syntax and semantics in a natural language such as English could be viewed as not essentially different from the relation between syntax and semantics in formal language such as the language of FOL. The framework he developed is known as "Montague's Universal Grammar" and its main components are:

- Model-Theory
- The principle of "Compositionality" which is due to Frege (1879).
- The λ-calculus which was invented by Church in the 1930's.
- Categorial Grammar which is due to Ajdukiewicz ('35) and Bar-Hill ('53). It's a (context free) Grammar based on functional application.

The novelty of Montagues' work was to apply them to natural language in a uniform framework.

2. Inference

We've said that an important ultimate question in Formal Semantics is "How do we infer some piece of information out of other?"

We have seen how to use logic to represent natural language input.

Therefore the question reduces to the question of "Which are the inference tools for the logic we used?".

For instance, a tool you already know is the "Tableaux Method" (**Logic** Course by Franconi).

In the **Computational Logic** course (by Martinenghi) you are studying the computational properties of this tool

3. Inference (Cont'd)

Inference plays a crucial role also at the discourse level. For instance,

"Jon has a rabbit. The tail is white and fluffy".

"The tail" of whom?

I know that most rabbits have a tail, hence Jon's rabbit has a tail. Therefore, I can interpret "the tail" to be of Jon's rabbit.

4. The Syntax-Semantics Interface

So far, we have spoken of syntax and semantics of natural language as two distinct and separate levels. However, as we know from our every-day use of NL these levels are tiedely connected.

Now, we will move to look at the interface between syntax and semantic.

Recall, from syntax we know that phrases are composed out of words, and from semantics we know that **meaning flows from the lexicon**.

Reference : L.T.F. Gamut "Logic, Language and Meaning", Vol. 2. The University of Chicago Press,1991. Chapter 4. (see library or ask copies to me)

4.1. Parallel vs. Non-parallel

We could build the meaning representation of an expression either

(a) in parallel with the construction of its syntactic structure, or

(b) after having built the syntactic analysis.

(a) is the method followed by most formal grammar frameworks as Categorial Grammar (CG), Head-Driven Phrase Structure Grammar (HPSG), Lexical Functional Grammar (LFG), Tree-Adjoining Grammar (TAG).

(b) is used by the Government and Binding Theory and the Minimalist Program (both due to Chomsky).

4.1.1. Advantages

The reasons for preferring the first approach are the following:

Psycholinguistic works suggest that human processing proceeds incrementally through the simultaneous application of syntactic, semantics, and phonological constraints to resolve syntactic ambiguity. (Though, note that these systems are models of linguistic competence rather than performance. Hence, these results could not provide direct support of either of the approaches.)

Computational approach requires a way to rule out a semantically ill-formed phrase as soon as it is encountered. Therefore, (a) offers a more efficient architecture for implementing constraint satisfaction. For instance,

1. The delegates met for an hour.
2. The committee met for an hour.
3. *The woman met for an hour.

The use of "met" as intransitive verb requires a subject denoting a plural entity.

4.2. Compositionally vs. Non-compositionally

- In **compositional** semantics theory the relation between the meaning of an expression and the meaning of its constituents is a **function**: to each distinct syntactic structure correspond a distinct interpretation.
- In **underspecification** theory this relation is systematic but it's **not a function**: an expression analyzed by a single syntactic structure can be associated with a set of alternative interpretations rather than with a unique semantic value. Sentences are assigned underspecified representation containing parameters whose value can be defined in several distinct ways. Constraints apply to filter the possible combinations of values for the set of parameters in such a schematic representation.

Reference: For underspecified semantics see BB1.

5. Lambda terms and DCG

We will look at the compositional approach to the syntax-semantics interface and build the meaning representation in parallel to the syntactic tree. This reduces to have a **rule-to-rule** system, i.e. each syntactic rule correspond to a semantic rule.

Syntactic Rule 1 $S \rightarrow NP\ VP$

Semantic Rule 1 If the logical form of the NP is α and the logical form of the VP is β then the logical form for the S is $\beta(\alpha)$.

Syntactic Rule 2 $VP \rightarrow TV\ \ NP$

Semantic Rule 2 If the logical form of the TV is α and the logical form of the NP is β then the logical form for the VP is $\alpha(\beta)$.

5.1. Augumenting DCG with terms

That can also be abbreviated as below where γ, α and β are the meaning representations of S, NP and VP, respectively.

$$(\gamma) \rightarrow NP(\alpha)\ VP(\beta)\ \gamma = \beta(\alpha)$$

This implies that lexical entries must now include semantic information. For instance, a way of writing this information is as below.

$$TV(\lambda x.\lambda y.wrote(y,x)) \rightarrow [wrote]$$

5.1.1. Exercise (a) Write the semantic rules for the following syntactic rules:

```
s --> np vp
vp --> iv
vp --> tv np
np --> det n
n --> adj n
n --> student
det --> a
adj --> tall
```

(b) apply these labeled rules to built the partial labeled parse trees for "A student" and "A tall student".

5.2. Quantified NP: class

In attempting to extend the technique of compositional semantics we run into problems with e.g. the rule for quantified noun phrases (QP).

QP should belong to the same category of noun phrases, as suggested by the substitution test or the coordination test. E.g.
pause

1. I will bring here **every student** and **Mary**.
2. I will bring here **John** and Mary.

5.3. Quantified NP: terms and syntactic rules

We have seen that the term of a quantifier like "every student" is $\lambda Y.\forall z.Student(z) \rightarrow Y(z)$, which is of type $(e \rightarrow t) \rightarrow t$. Hence the sentence,

Every student left.

is obtained by applying the quantified noun phrase to the verb. In other words, if "Everybody" is of category NP we need the rule below:

$$S(\gamma) \rightarrow NP(\alpha)\ VP(\beta)\ \gamma = \alpha(\beta)$$

5.4. Quantified NP and Proper Nouns

This has brought semanticists to change the meaning representation of the noun phrase too, since they have to be of the same "sort". E.g, "John" could be represented as

$$\lambda X.X(john)$$

a function of the same type as the quantified NP, i.e. $(e \rightarrow t) \rightarrow t$.

ie., what would be its set-theoretical meaning?

6. Categorial Grammar

- **Who:** Lesniewski (1929), Ajdukiewicz (1935), Bar-Hillel (1953).
- **Aim:** To build a language recognition device.
- **How:** Linguistic strings are seen as the result of concatenation obtained by means of syntactic rules starting from the categories assigned to lexical items. The grammar is known as Classical Categorial Grammar (CG).
- **Connection with Type Theory:** The syntax of type theory closely resembles the one of categorial grammar. The links between types (and lambda terms) with models, and types (and lambda terms) with syntactic categories, gives an interesting framework in which syntax and semantic are strictly related. (We will come back on this later.)

Categories: Given a set of basic categories ATOM, the set of categories CAT is the smallest set such that:

$$\mathsf{CAT} := \mathsf{ATOM} \mid \mathsf{CAT}\backslash\mathsf{CAT} \mid \mathsf{CAT}/\mathsf{CAT}$$

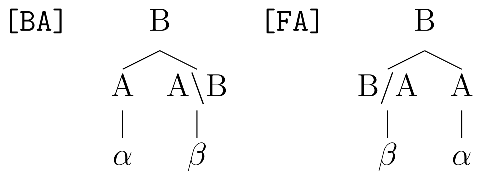
7. CG: Syntactic Rules

Categories can be composed by means of the syntactic rules below

[BA] If α is an expression of category A, and β is an expression of category $A\backslash B$, then $\alpha\beta$ is an expression of category B.

[FA] If α is an expression of category A, and β is an expression of category B/A, then $\beta\alpha$ is an expression of category B.

where [FA] and [BA] stand for Forward and Backward Application, respectively.



8. CG Lexicon: Toy Fragment

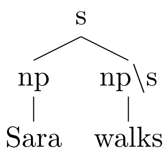
Let ATOM be $\{n, s, np\}$ (for nouns, sentences and noun phrases, respectively) and LEX as given below. Recall PSG rules: $np \rightarrow det\ n, s \rightarrow np\ vp, vp \rightarrow v\ np\ \ldots$

Lexicon

Sara	np	the	np/n
student	n	walks	$np\backslash s$
wrote	$(np\backslash s)/np$		

Sara walks $\in s$? $\rightsquigarrow$ $\underbrace{np}_{\text{Sara}}, \underbrace{np\backslash s}_{\text{walks}} \in s$? Yes

simply [BA]



9. Classical Categorial Grammar

Alternatively the rules can be thought of as Modus Ponens rules and can be written as below.

$$B/A, A \Rightarrow B \qquad \mathrm{MP_r}$$
$$A, A\backslash B \Rightarrow B \qquad \mathrm{MP_l}$$

$$\frac{B/A \quad A}{B}\ (\mathrm{MP_r}) \qquad \frac{A \quad A\backslash B}{B}\ (\mathrm{MP_l})$$

10. Classical Categorial Grammar. Examples

Given $\mathsf{ATOM} = \{np, s, n\}$, we can build the following lexicon:

Lexicon

John, Mary	$\in$	np	the	$\in$	np/n
student	$\in$	n			
walks	$\in$	$np\backslash s$			
sees	$\in$	$(np\backslash s)/np$			

Analysis

John walks $\in s$? $\quad\rightsquigarrow\quad np, np\backslash s \Rightarrow s$? $\quad$ Yes

$$\frac{np \quad np\backslash s}{s}\,(\mathrm{MP_l})$$

John sees Mary $\in s$? $\quad\rightsquigarrow\quad np, (np\backslash s)/np, np \Rightarrow s$? $\quad$ Yes

$$\frac{np \qquad \dfrac{(np\backslash s)/np \quad np}{np\backslash s}\,(\mathrm{MP_r})}{s}\,(\mathrm{MP_l})$$

10.1. Relative Pronoun

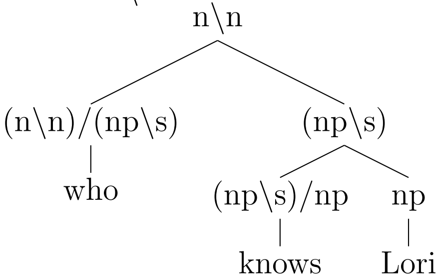
Question Which would be the syntactic category of a relative pronoun in subject position? E.g. "the student **who** knows Lori"

[the [[student]$_n$ [who [knows Lori]$_{(np\backslash s)}$]$_?$]$_n$

who knows Lori $\in n\backslash n$? $\leadsto$

$$(n\backslash n)/(np\backslash s), (np\backslash s)/np, np \Rightarrow n\backslash n?$$

$$\dfrac{\dfrac{\text{who}}{(n\backslash n)/(np\backslash s)} \quad \dfrac{\dfrac{\text{knows}}{(np\backslash s)/np} \quad \dfrac{\text{Lori}}{np}}{np\backslash s}\,(\text{MP}_\text{r})}{n\backslash n}\,(\text{MP}_\text{r})$$



10.2. CFG and CG

Below is an example of a simple CFG and an equivalent CG:

CFG

```
S --> NP VP
VP --> TV NP
N --> Adj N

Lexicon:
Adj --> poor
NP --> john
TV --> kisses
```

CG Lexicon:

John:	np
kisses:	$(np\backslash s)/np$
poor:	n/n

11. CG: syntax-semantics interface

Summing up, CG specifies a language by describing the **combinatorial possibilities of its lexical items** directly, without the mediation of phrase-structure rules. Consequently, two grammars in the same system differ only in the lexicon.

The **close relation between the syntax and semantics** comes from the fact that the two syntactic rules are application of a functor category to its argument that corresponds to functional application of the lambda calculus.

We have to make sure that the lexical items are associated with **semantic terms** which correspond to the **syntactic categories**.

11.1. Mapping: types-categories

To set up the form-meaning correspondence, it is useful to build a language of semantic types in parallel to the syntactic type language.

Definition 11.1 (Types) Given a non-empty set of basic types Base, the set of types TYPE is the smallest set such that

i. $\mathsf{Base} \subseteq \mathsf{TYPE}$;
ii. $(a \rightarrow b) \in \mathsf{TYPE}$, if a and $b \in \mathsf{TYPE}$.

Note that this definition closely resembles the one of the syntactic categories of CG. The only difference is the lack of directionality of the functional type (a, b). A function mapping the syntactic categories into TYPE can be given as follows.

Definition 11.2 (Categories and Types) *Let us define a function* $\mathtt{type} : \mathsf{CAT} \rightarrow \mathsf{TYPE}$ *which maps syntactic categories to semantic types.*

$$\mathtt{type}(np) = e; \qquad \mathtt{type}(A/B) = (\mathtt{type}(B) \rightarrow \mathtt{type}(A));$$
$$\mathtt{type}(s) = t; \qquad \mathtt{type}(B\backslash A) = (\mathtt{type}(B) \rightarrow \mathtt{type}(A));$$
$$\mathtt{type}(n) = (e \rightarrow t).$$

11.2. CG: categories and terms

Modus ponens corresponds to functional application.

$$\frac{B/A : t \quad A : r}{B : t(r)}\ (\mathrm{MP_r}) \qquad\qquad \frac{A : r \quad A\backslash B : t}{B : t(r)}\ (\mathrm{MP_l})$$

Example

$$\frac{np : \texttt{john} \quad np\backslash s : \texttt{walk}}{s : \texttt{walk}(\texttt{john})}\ (\mathrm{MP_l})$$

$$np\backslash s : \lambda x.\texttt{walk}(x) \qquad (\lambda x.\texttt{walk}(x))(\texttt{john}) \rightsquigarrow_{\lambda-\mathrm{conv.}} \texttt{walk}(\texttt{john})$$

$$\frac{np : \texttt{john} \qquad \dfrac{(np\backslash s)/np : \texttt{know} \quad np : \texttt{mary}}{np\backslash s : \texttt{know}(\texttt{mary})}\ (\mathrm{MP_r})}{s : (\texttt{know}(\texttt{mary}))(\texttt{john})}\ (\mathrm{MP_l})$$

12. Practical things

http://www.inf.unibz.it/~bernardi/Courses/CompLing/08-09.html