1. Recall: Constituents and Assembly

Let's go back to the points where FOL fails, i.e. constituent representation and assembly. The λ-calculus succeeds in both:

Constituents: each constituent is represented by a lambda term.

John: $\mathtt{j}$ knows: $\lambda xy.(\mathtt{know}(x))(y)$ read john: $\lambda y.\mathtt{know}(y, \mathtt{j})$

Assembly: function application $(\alpha(\beta))$ and abstraction $(\lambda x.\alpha[x])$ capture composition and decomposition of meaning representations.

2. Lambda terms and CFG

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 $\alpha(\beta)$.

2.1. Augumenting CFG with terms

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

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

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

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

2.1.1. 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 build the partial labeled parse trees for “A student” and “A tall student”.

2.2. 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.

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

2.3. 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:

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

2.4. 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. $(e \to t) \to t$.

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

3. 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:

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

4. CG: Syntactic Rules

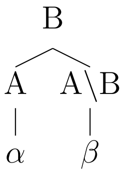
Categories can be composed by means of the syntactic rules below

`[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.

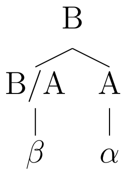
`[FA]` If α is an expression of category A, and β is an expression of category B/A, then $\beta\alpha$ is an expression of category B.

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

`[BA]`



`[FA]`



5. CG Lexicon: Toy Fragment

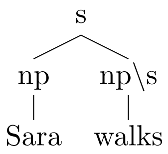
Let ATOM be $\{n, s, np\}$ (for nouns, sentences and noun phrases, respectively) and LEX as given below. Recall PSG rules: $np \rightarrow det\ n$, $s \rightarrow np\ vp$, $vp \rightarrow v\ np \ldots$

Lexicon

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

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

simply [BA]



6. Classical Categorial Grammar

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

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

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

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

7. 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 $\in s$? $\rightsquigarrow$ $np, np\backslash s \Rightarrow s$? Yes

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

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

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

7.1. Relative Pronoun

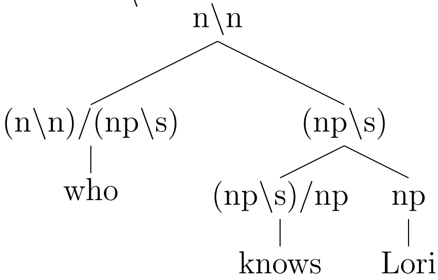
Question Which would be the syntactic category of a relative pronoun in subject position? E.g. “the student **who** knows Lori”

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

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

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

$$\cfrac{\cfrac{\text{who}}{(n\backslash n)/(np\backslash s)} \qquad \cfrac{\cfrac{\text{knows}}{(np\backslash s)/np} \quad \cfrac{\text{Lori}}{np}}{np\backslash s}\,(\mathrm{MP_r})}{n\backslash n}\,(\mathrm{MP_r})$$



7.2. 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

8. Next Time

We will look at:

- how CG relates to lambda terms
- what is needed in a Formal Grammar to exploit also abstraction.