Computational Linguistics: Categorial Grammar

Raffaella Bernardi

University of Trento

1. Last time and today

Last time we have:

- practiced on CFG and
- given an historical overview of FG.

Today, we will look at

- TAG and
- CG

2. Tree Adjoining Grammar (TAG)

- **Who:** Aravind Joshi (1969).
- **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 trees assigned to lexical items. The grammar is known as Tree Adjoining Grammar (TAG).
- `http://www.cis.upenn.edu/~xtag/`

2.1. CFG and TAG

```
CFG:



S -->  NP VP      NP --> Harry        ADV --> passionately
VP --> V NP       NP --> peanuts
VP --> VP ADV     V --> likes

TAG:

a1   S                  a2    NP            a3 NP
   /   \                      |                |
 NP|     VP                   peanuts          Harry
       /    \
      V     NP |
      |
   likes
```

2.2. TAG rules

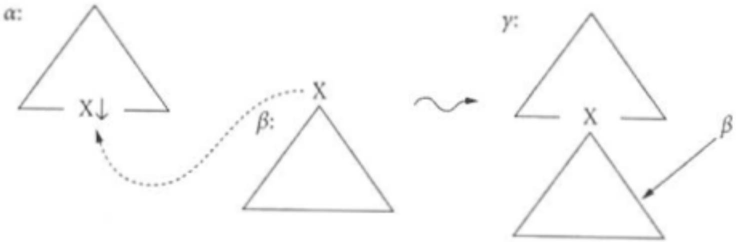


Fig. 26.2 Substitution

2.3. Example

Try to apply the substitution rules to the entries given below:

```
a1   S                  a2    NP             a3 NP
   /   \                      |                 |
 NP|     VP                   peanuts           Harry
     /    \
     V     NP |
     |
   likes
```

Do you think this rule is going to be enough?

2.4. Example

"Harry thinks Bill likes John"

what's the entry for "thinks"?

```
     S
   /   \
NP|     VP
       /  \
      V    S|
      |
    think
```

And what about the sentence "Who does Harry think Bill likes?"

2.5. Example

To account for gaps, new elementary trees are assigned to e.g. TV:

```
           S
         /    \
NP(wh)|       S
            /   \
          NP|    VP
                /  \
               V    NP|
               |     |
             likes empty
```

2.6. Adjunction

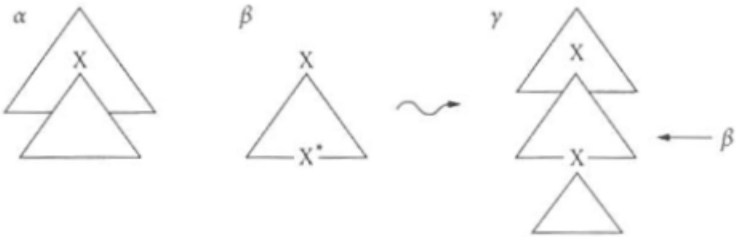
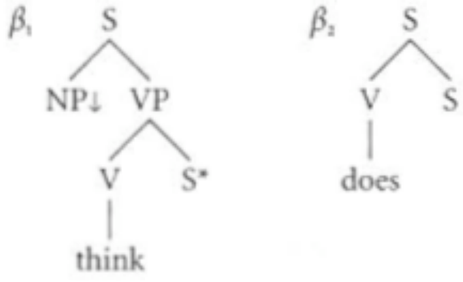


Fig. 26.5 Adjoining

The lexical entries “does” and “think” carry the special marker:



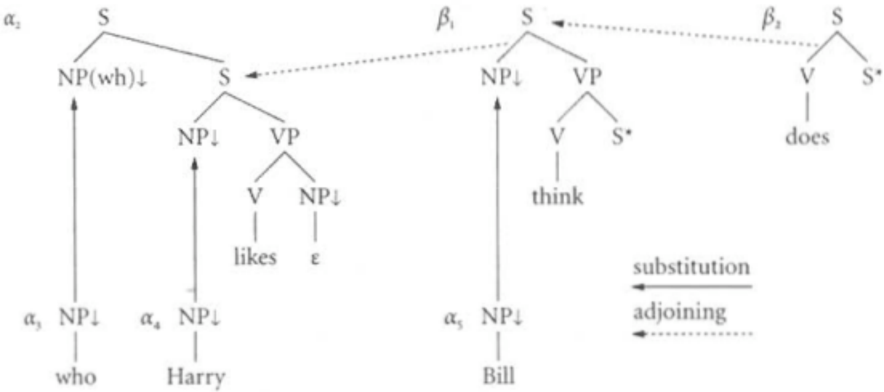


Fig. 26.9 LTAG derivation for *who does Bill think Harry likes*

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).

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

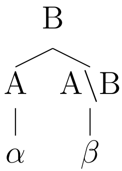
Categories can be composed by means of the syntactic rules below

`[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.

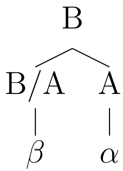
`[FA]` If α is an expression of category A, and β is an expression of category B/A, then $\beta\alpha$ is an expression of category B.

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

`[BA]`



`[FA]`



5. CG Lexicon: Toy Fragment

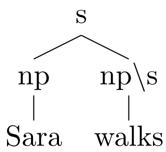
Let ATOM be $\{n, s, np\}$ (for nouns, sentences and noun phrases, respectively) and LEX as given below. Recall CFG rules: $np \rightarrow det\ n, s \rightarrow np\ vp, vp \rightarrow v\ np \ldots$

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]



6. Classical Categorial Grammar

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

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

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

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

7. Classical Categorial Grammar. Examples

Given 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

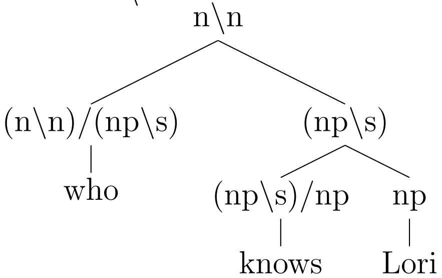
Question Which would be the syntactic category of a relative pronoun in subject position? E.g. “the student **who** knows Lori”

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

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

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

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



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. Logic Grammar

- **Aim:** To define the logic behind CG.
- **How:** Considering categories as formulae; $\backslash, /$ as logic connectives.
- **Who:** Jim Lambek [1958]

8.1. Lambek Calculi

In the Lambek Calculus the connectives are $\backslash$ and $/$ (that behave like the $\rightarrow$ of PL except for their directionality aspect.)

Therefore, in the Lambek Calculus besides the elimination rules of $\backslash, /$ (that we saw in CG) we have their introduction rules.

$$\frac{B/A \quad A}{B}\ /\mathrm{E} \qquad \frac{A \quad A\backslash B}{B}\ \backslash\mathrm{E}$$

$$\frac{\begin{array}{c}[A]^i\\ \vdots\\ B\end{array}}{B/A}\ /\mathrm{I}^i \qquad \frac{\begin{array}{c}[A]^i\\ \vdots\\ B\end{array}}{A\backslash B}\ \backslash\mathrm{I}^i$$

Remark The introduction rules do not give us a way to distinguish the directionality of the slashes.

8.2. Alternative Notation (Sequents)

Let A, B, C stand for logic formulae (e.g. np, $np\backslash s$, $(np\backslash s)\backslash(np\backslash s)\ldots$) i.e. the categories of CG

Let Γ, Σ, Δ stand for structures (built recursively from the logical formulae by means of the $\circ$ connective) –e.g. $np \circ np\backslash s$ is a structure. $\mathtt{STRUCT} := \mathsf{CAT}, \mathtt{STRUCT} \circ \mathtt{STRUCT}$

$\Sigma \vdash A$ means that (the logic formula) A derives from (the structure) Σ (e.g. $np \circ np\backslash s \vdash s$).

$$A \vdash A$$

$$\frac{\Delta \vdash B/A \quad \Gamma \vdash A}{\Delta \circ \Gamma \vdash B}\ (/E) \qquad \frac{\Gamma \vdash A \quad \Delta \vdash A\backslash B}{\Gamma \circ \Delta \vdash B}\ (\backslash E)$$

$$\frac{\Delta \circ A \vdash B}{\Delta \vdash B/A}\ (/I) \qquad \frac{A \circ \Delta \vdash B}{\Delta \vdash A\backslash B}\ (\backslash I)$$

9. Lambek calculus. Elimination rule

$$\frac{np \vdash np \qquad np\backslash s \vdash np\backslash s}{\underbrace{np}_{\text{sara}} \circ \underbrace{np\backslash s}_{\text{walks}} \vdash s}$$

$$\frac{np \vdash np \qquad \dfrac{(np\backslash s)/np \vdash (np\backslash s)/np \qquad np \vdash np}{(np\backslash s)/np \circ np \vdash np\backslash s}}{\underbrace{np}_{\text{sara}} \circ (\underbrace{(np\backslash s)/np}_{\text{knows}} \circ \underbrace{np}_{\text{mary}}) \vdash s}$$

9.1. Lambek calculus. Subject relative pronoun

$$\underbrace{\text{The student who [[\ldots] knows Mary]}_s}_{np}\ \underbrace{\text{left}}_{np\backslash s}$$

$$\dfrac{(n\backslash n)/(np\backslash s) \vdash (n\backslash n)/(np\backslash s) \qquad \dfrac{(np\backslash s)/np \vdash (np\backslash s)/np \quad np \vdash np}{(np\backslash s)/np \circ np \vdash np\backslash s}}{\underbrace{(n\backslash n)/(np\backslash s)}_{\text{who}} \circ (\underbrace{(np\backslash s)/np}_{\text{knows}} \circ \underbrace{np}_{\text{mary}}) \vdash n\backslash n}$$

Exercise: Try to do the same for relative pronoun in object position. e.g. the student who Mary met (i.e. prove that it is of category np. Which should be the category for a relative pronoun (e.g. who) that plays the role of an object?

10. Lambek calculus. Introduction rule

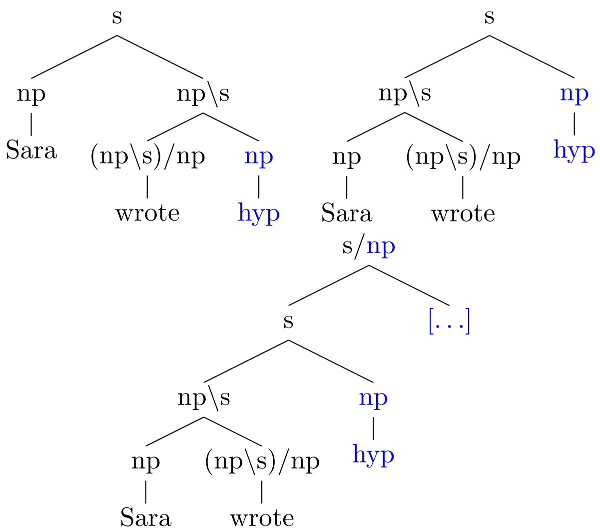
Note, below for simplicity, I abbreviate structures with the corresponding linguistic structures.

$$\underbrace{\text{The book which [Sara wrote [...]]}_s}_{np}\ \underbrace{\text{is interesting}}_{np\backslash s}.$$

$$\dfrac{\text{which} \vdash (n\backslash n)/(s/np) \qquad \dfrac{\dfrac{\text{Sara} \vdash np \qquad \dfrac{\text{wrote} \vdash (np\backslash s)/np \quad [np \vdash np]^1}{\text{wrote } np \vdash np\backslash s}\,(/\text{E})}{\text{Sara wrote } np \vdash s}\,(\backslash\text{E})}{\text{Sara wrote} \vdash s/np}\,(/\text{I})^1}{\text{which Sara wrote} \vdash n\backslash n}\,(/\text{E})$$

Introduction rules accounted for extraction.

11. Extraction: Right-branch (tree)



12. CCG

A well known version of CG is CCG (Combinatory Categorial Grammar) developed by Mark Steedman (Edinburgh University).

- CCG Bank
- C&C parser
- C&C parser together with Boxer (MR builder).

Link to some softwares: http://groups.inf.ed.ac.uk/ccg/software.html

13. (Recall) Generative Power and Complexity of FGs

Every (formal) grammar generates a unique language. However, one language can be generated by several different (formal) grammars.

Formal grammars differ with respect to their **generative power**:

One grammar is of a greater generative power than another if it can recognize a language that the other cannot recognize.

Two grammars are said to be

- **weakly** equivalent if they generate the same string language.
- **strongly** equivalent if they generate both the same string language and the same tree language.

13.1. DG, CG, CTL, CCG, and TAG

- DG: Gross (1964)(p.49) claimed that the dependency languages are **exactly** the context-free languages. This claim turned out to be a mistake, and now there is new interest in DG. (Used in QA)
- CG: Chomsky (1963) conjectured that **Lambek calculi** were also **context-free**. This conjecture was proved by Pentus and Buszkowski in 1997.
- TAG and CCG: have been proved to be Mildly Context Free.
- CTL has been proved to be Mildly Sensitive (Moot), or Context Sensitive (Moot) or Turing Complete (Carpenter), accordingly to the structural rules allowed.

14. Next steps

Next time, we will practice with CG rules

- Wednesday we will practice with CG rules.
- Thu. we will introduce Formal Semantics
- Monday. we will look at Distributional Semantics
- Wed. we will look at Compositional DS
- Thur. we will look at the syntax-semantics interface in CFG and CG.
- Recall: 03.04.2017 SAMPLE EXAM