

Notes (12/5/98):

Categorial Grammars

Atomic categories; For example, N, S

Complex categories: If X and Y are categories then X/Y and $X\backslash Y$ are also categories.

Examples:

$(S\backslash NP)/NP$: for transitive verbs

$S\backslash NP$: for intransitive verbs

$(S\backslash NP)/NP(\text{directobject})/NP(\text{indirectobject})$: for ditransitive verbs

$(S\backslash NP)\backslash(S\backslash NP)$: for adverbs

N/N : for adjectives

NP/N : for determiners

Reductions:

Forward function application: $X/Y \ Y \rightarrow X$

Backward function application: $Y \ X\backslash Y \rightarrow X$

Function composition: $X/Y \ Y/Z \rightarrow X/Z$

(Similarly for the backslash)

Mixed composition: $X/Y \ Y\backslash Z \rightarrow X\backslash Z$

Type raising $X \rightarrow Y/(Y\backslash X)$

Some derivations:

John likes apples

NP (S\NP)/NP NP

(S\NP)

S

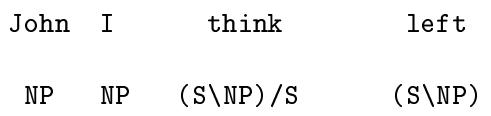
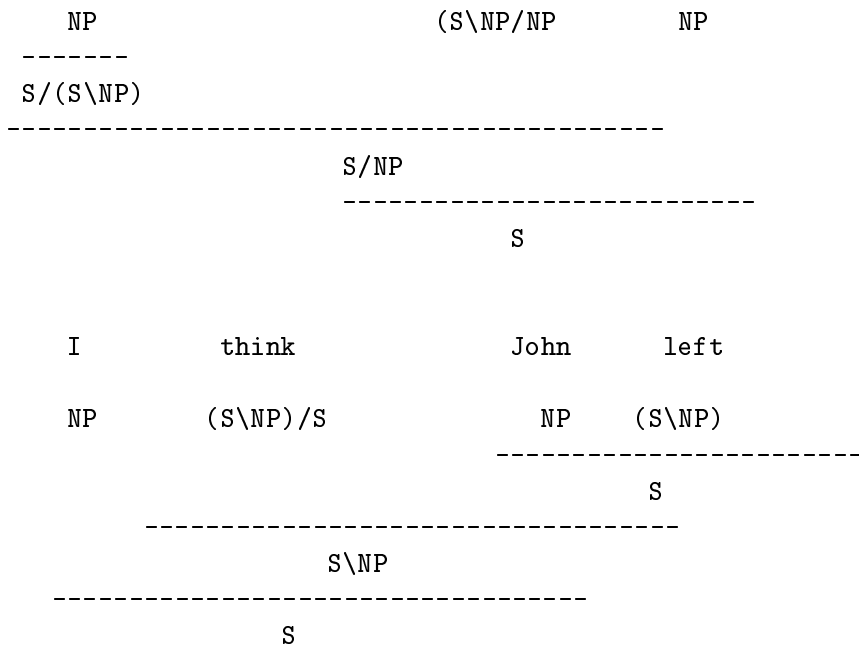
John left abruptly

NP (S\NP) (S\NP)\(S\NP)

S\NP

S

John likes apples



Continue this derivation:

Hint-- type raise 'I' combine 'I' and 'think' and then with 'left' etc.

Categorial grammars with function application only are (weakly) equivalent to Context-free grammars. This follows from the fact for any CFG we can construct an equivalent CFG in the Greibach Normal Form, i.e., a grammar whose rules are of the form

$$A \rightarrow aB_1B_2B_3...B_n$$

$$A \rightarrow a$$

These rules correspond to categorial assignments as follows:

$$a : A/B_n/B_{n-1}/.../B_2/B_1$$

$$a : A$$

Categorial grammars with more complex compositions can be shown to be more powerful than CFGs.

Coordination:

One major advantage of a categorial grammar is that it can allow flexible constituency. For example, as we have seen before we obtain two analyses for *John likes apples*, one in which *likes apples* is a constituent and in the other case *John apples* is a constituent. This permits both

John likes apples and hates pears

John likes and Bill hates pears

A problem: Show that both *John* and *Harry thinks Mary* can have the same type. Hint: Type raise *John*, type-raise *Harry* and *Mary* and then combine *Harry* and *thinks* and then with *Mary*. What is the problem? What is a possible solution?