Ellipsis and Reduced Registers

Day 1

Syntax crash-course #1:

(talking about) phrase structures

Background in formal syntax?

Constituency
- Syntactic phrases
- Syntactic heads
- Projection

Structural dependencies
- Syntactic case
- Referential binding
- Negative polarity items

Syntactic trees
- X'-theory
- C-command
- Dominance
- Complements
- Specifiers

Syntactic movement
- Islands

The Y-model of grammar

What is syntax?

"The arrangement of words and phrases to create well-formed sentences in a language." (Oxford English Dictionary)

What is a syntactic analysis?

- A description of all the grammatical and ungrammatical sentences in a particular language L.

(English, German, Turkish, etc.)

Listing?

[1] Trump doesn’t like kimchi.
[2] Fred should ask Julie out.
[3] Listing sentences is very dull.
[4] Did you watch the match last night?

Problem: Endless sentences

John thinks that Mary knows that Pete claims that Lucy is sure that Fred believes that …

… this sentence is as endless as an endless rainbow that…

Language is **generative** – it generates sentences

Conclusion: syntax is a set of generative rules.

The job of syntacticians:

Create the simplest possible set of rules for generating the grammatical and ungrammatical sentences in a language L.

What's the input to syntactic rules?

[1] Syntactic categories (parts of speech)

nouns = John, cat, beauty, herself, …

verbs = fall, kiss, solidify, …

prepositions = in, after, between, …

determiners = a, the, some, several, …

…

What's the input to syntactic rules?

Potential rule:
A tensed V must be immediately preceded by an N

(3) Bill laughed. 

(4) Sally scares John. 

(5) * Quickly scares John. 

(6) The house with ghosts inside scares John. 

What’s the input to syntactic rules?

(6) The house with ghosts inside scares John.

What's the input to syntactic rules?

(6) The house with ghosts inside scares John.

Noun Phrase (NP)

(actually "DP")

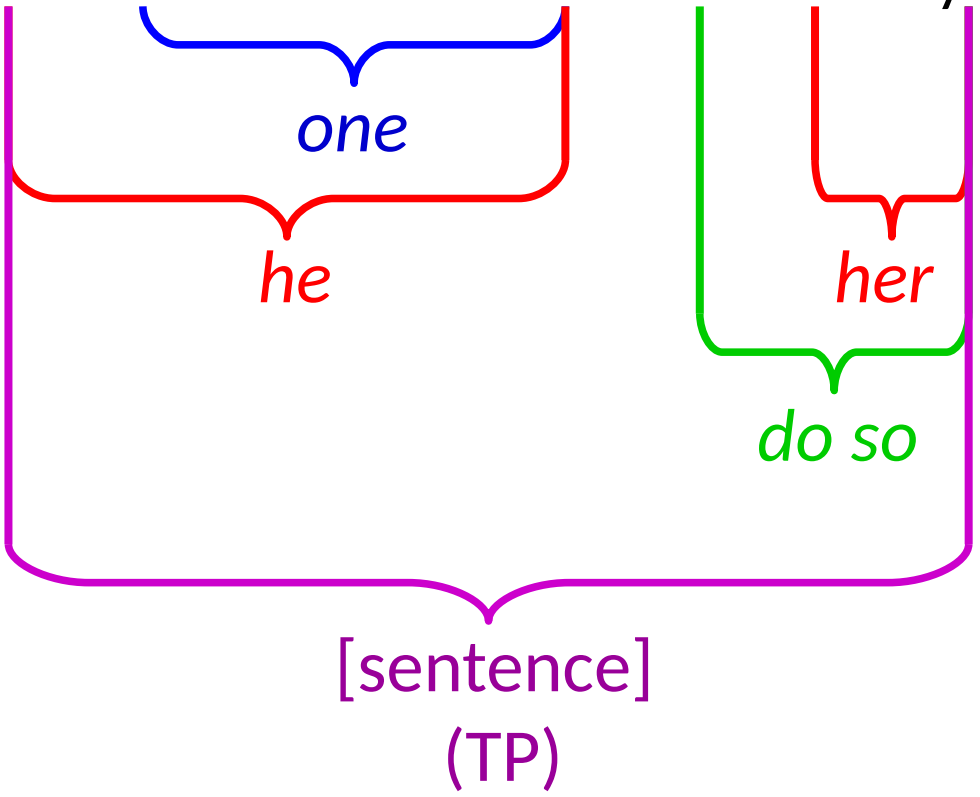
- Parts of speech (e.g. nouns) can **project** their category
- The maximal extent of their projection is a **phrase**
- The projecting item is the **head** of the phrase

What's the input to syntactic rules?

Diagnosing phrases:

- Can chunks of words be replaced with pronominal elements?

(7) This handsome man will kiss Mary.



NP = one

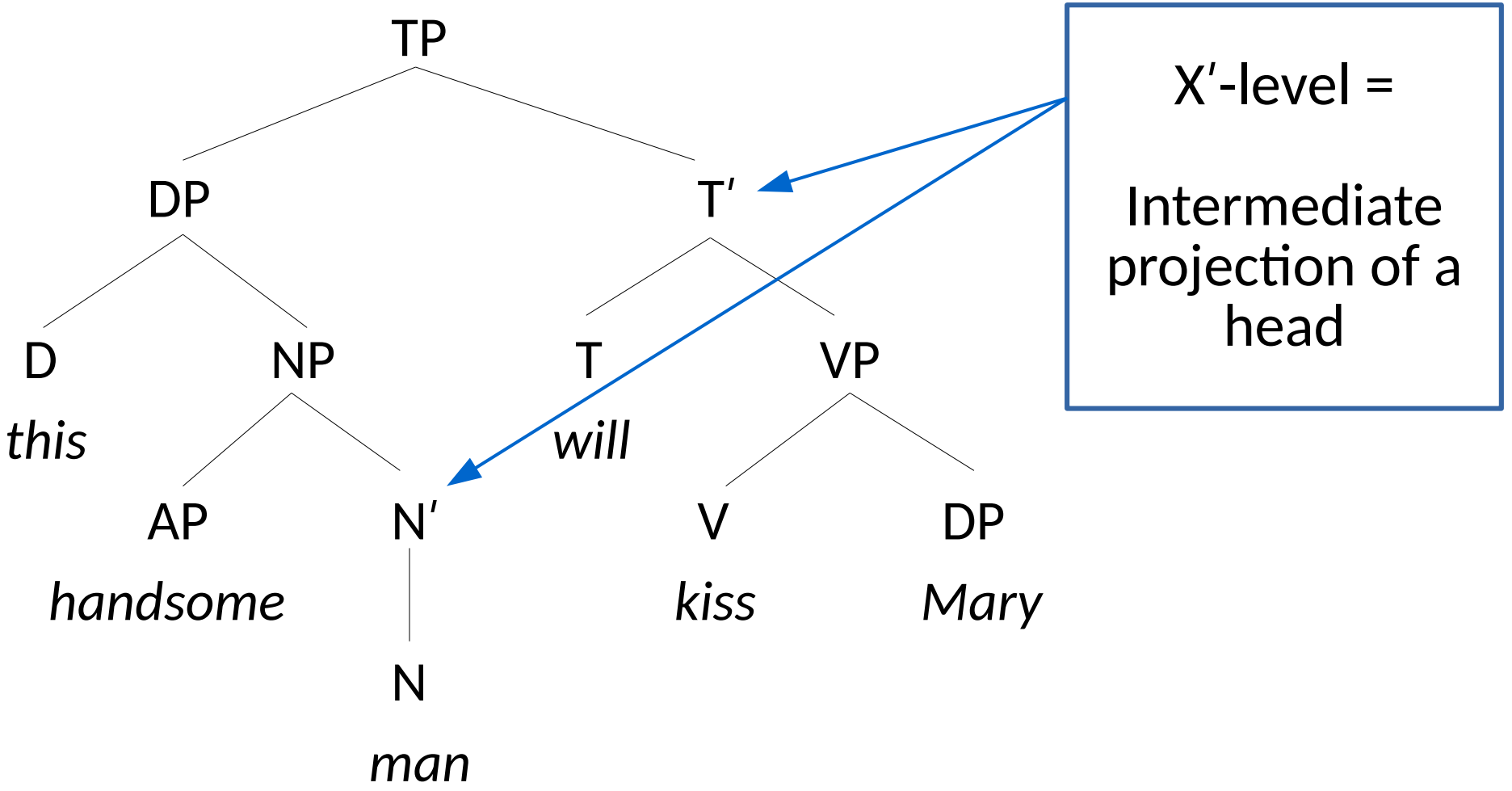
DP = he, she, him …

VP = do so

What's the input to syntactic rules?

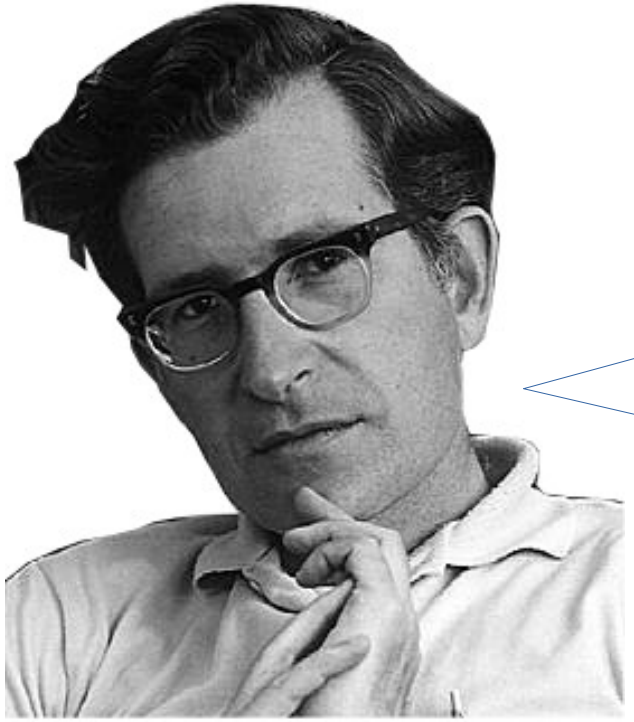
(7) [$_{TP}$ [$_{DP}$ This [$_{NP}$ handsome man]] will [$_{VP}$ kiss [$_{DP}$ Mary]]].

(8)



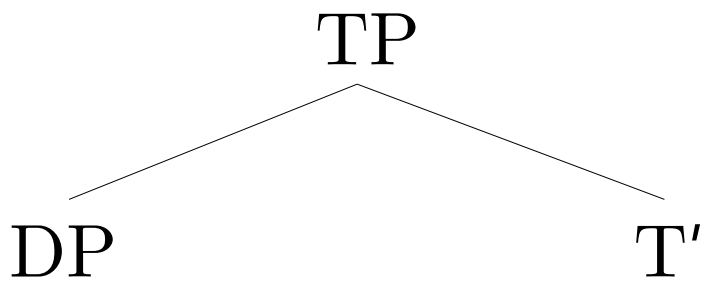
Formulating syntactic rules (old style)

Noam Chomsky (1959):



We can describe a language in terms of ***Phrase Structure*** rules

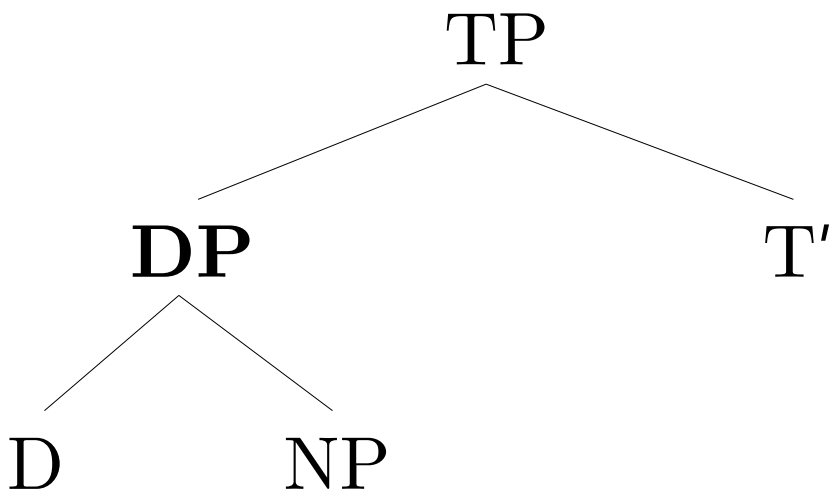
Phrase Structure rules for a fragment of English



$$
\begin{array}{ll}
\underline{\text{TP}} & \rightarrow \text{DP} + \text{T}' \\
\text{T}' & \rightarrow \text{T} + \text{VP} \\
\text{DP} & \rightarrow \text{D} + \text{NP} \\
\text{VP} & \rightarrow \text{V} + \text{DP}
\end{array}
$$

the dog will chase the cat

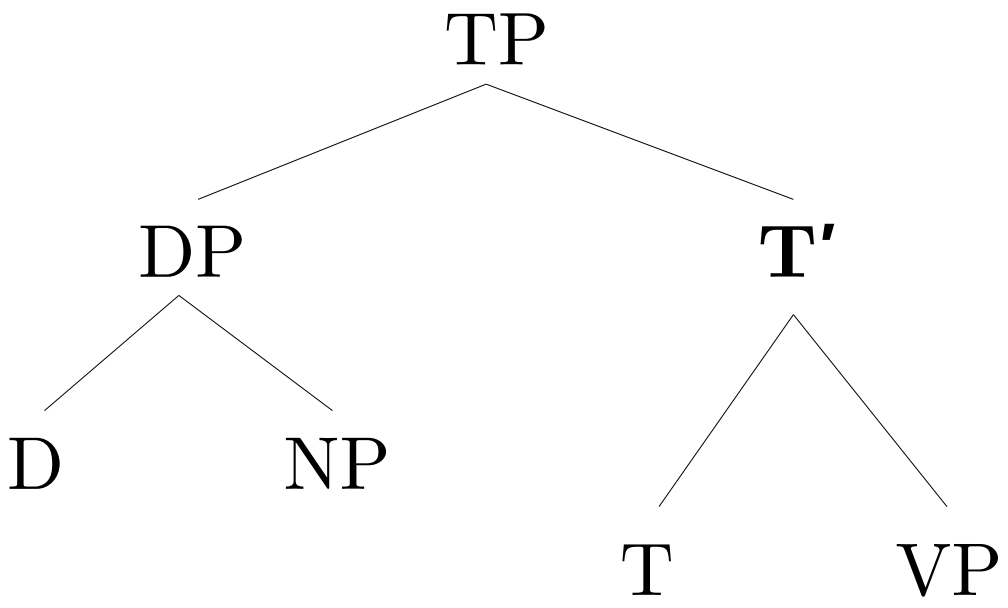
Phrase Structure rules for a fragment of English



$$
\begin{aligned}
\underline{\text{TP}} &\rightarrow \text{DP} + \text{T}' \\
\text{T}' &\rightarrow \text{T} + \text{VP} \\
\text{DP} &\rightarrow \text{D} + \text{NP} \\
\text{VP} &\rightarrow \text{V} + \text{DP}
\end{aligned}
$$

the dog will chase the cat

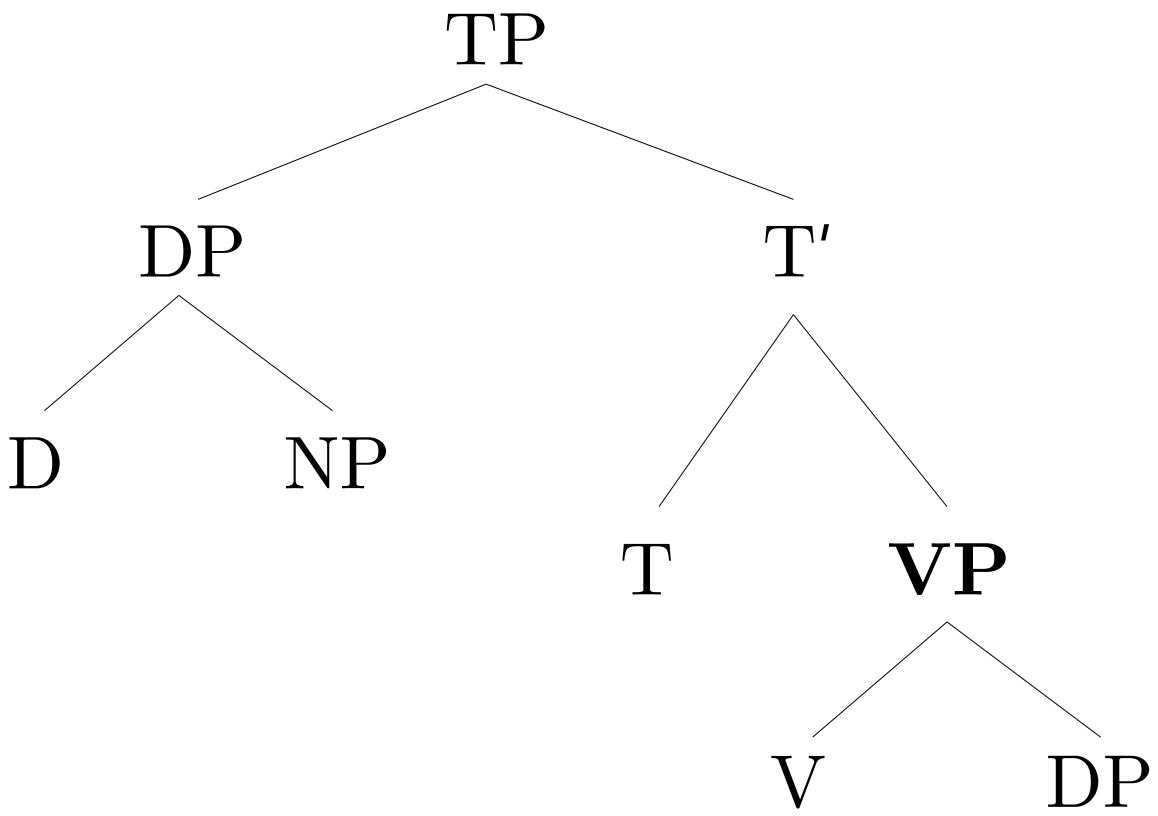
Phrase Structure rules for a fragment of English



$$\begin{aligned} \underline{\text{TP}} &\rightarrow \text{DP} + \text{T}' \\ \text{T}' &\rightarrow \text{T} + \text{VP} \\ \text{DP} &\rightarrow \text{D} + \text{NP} \\ \text{VP} &\rightarrow \text{V} + \text{DP} \end{aligned}$$

the dog will chase the cat

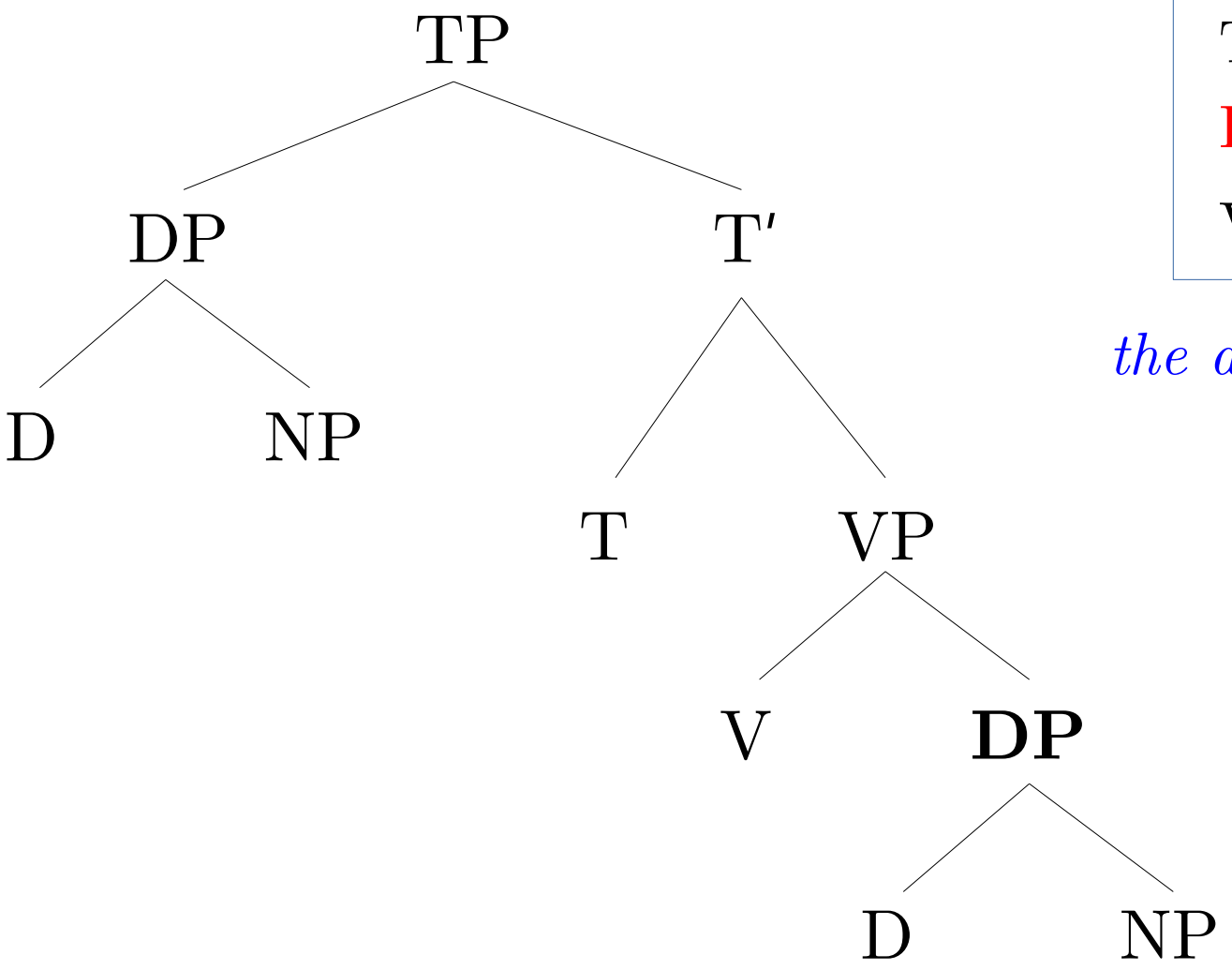
Phrase Structure rules for a fragment of English



$$\begin{aligned} \underline{\text{TP}} &\rightarrow \text{DP} + \text{T}' \\ \text{T}' &\rightarrow \text{T} + \text{VP} \\ \text{DP} &\rightarrow \text{D} + \text{NP} \\ \text{VP} &\rightarrow \text{V} + \text{DP} \end{aligned}$$

the dog will chase the cat

Phrase Structure rules for a fragment of English

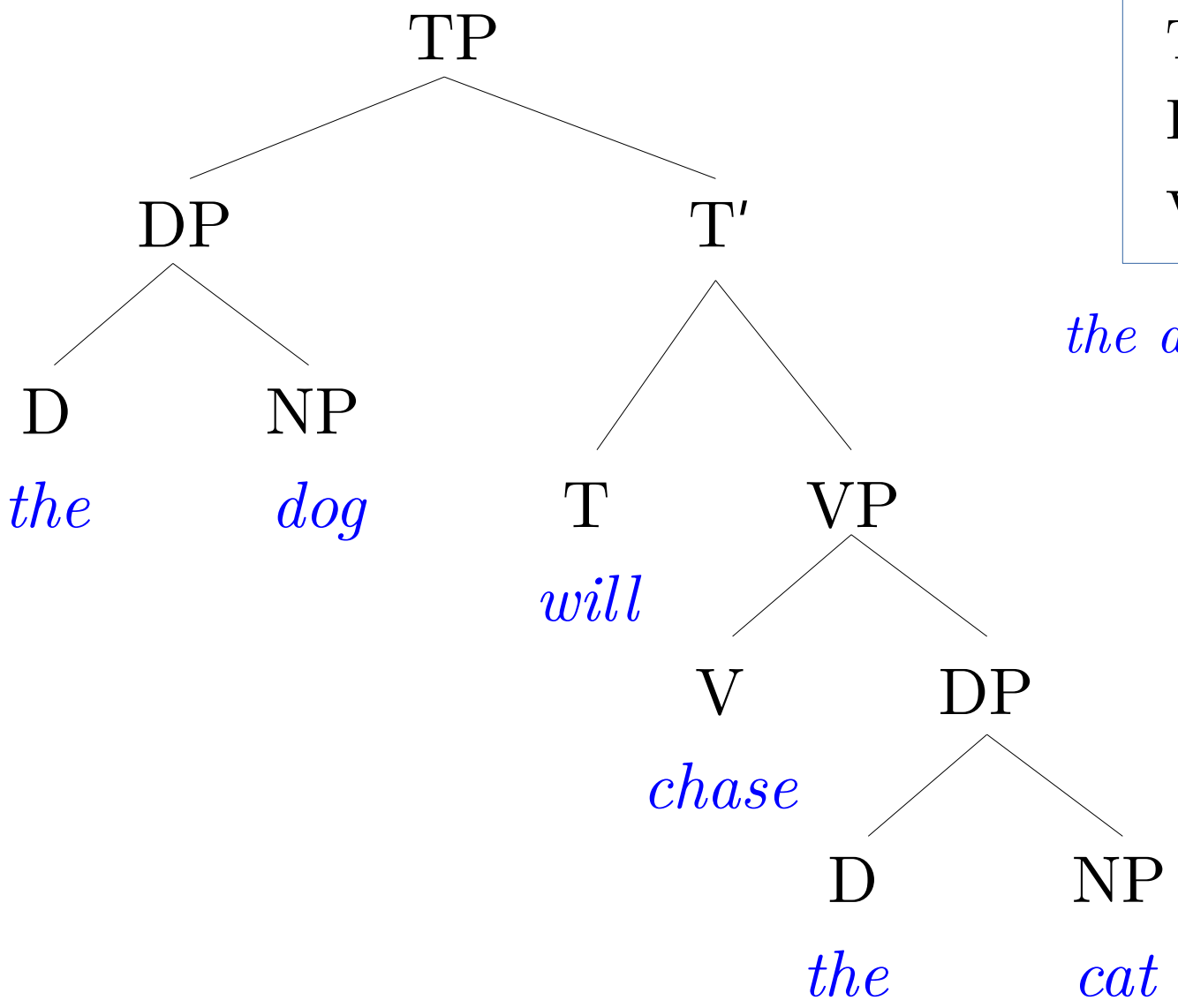


$\underline{\text{TP}}$	$\rightarrow$ DP + T′
T′	$\rightarrow$ T + VP
DP	$\rightarrow$ D + NP
VP	$\rightarrow$ V + DP

the dog will chase the cat

Phrase Structure rules for a fragment of English

$$
\begin{aligned}
\underline{\text{TP}} &\rightarrow \text{DP} + \text{T}' \\
\text{T}' &\rightarrow \text{T} + \text{VP} \\
\text{DP} &\rightarrow \text{D} + \text{NP} \\
\text{VP} &\rightarrow \text{V} + \text{DP}
\end{aligned}
$$



the dog will chase the cat

- If we develop the correct Phrase Structure rules for an entire language,we can generate all and only the grammatical sentences for that language.

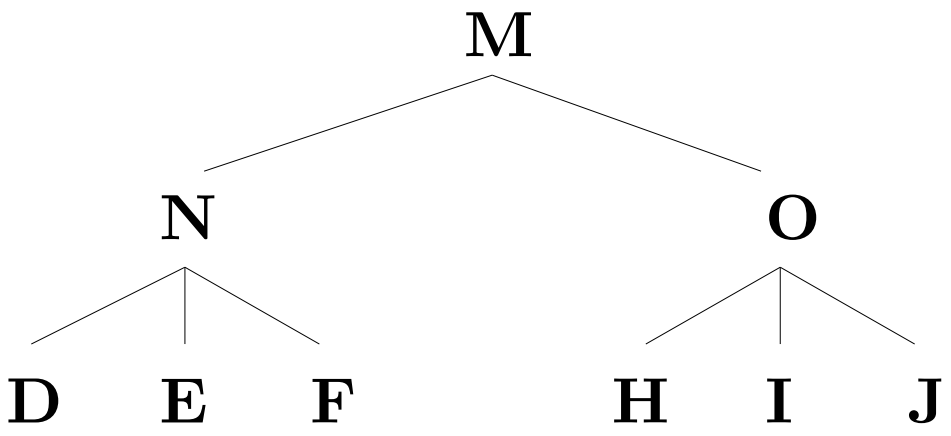
- We therefore fulfill the goal of syntactic analysis:

“What is a syntactic analysis?

- A description of all the grammatical and ungrammatical sentences in a particular language L.

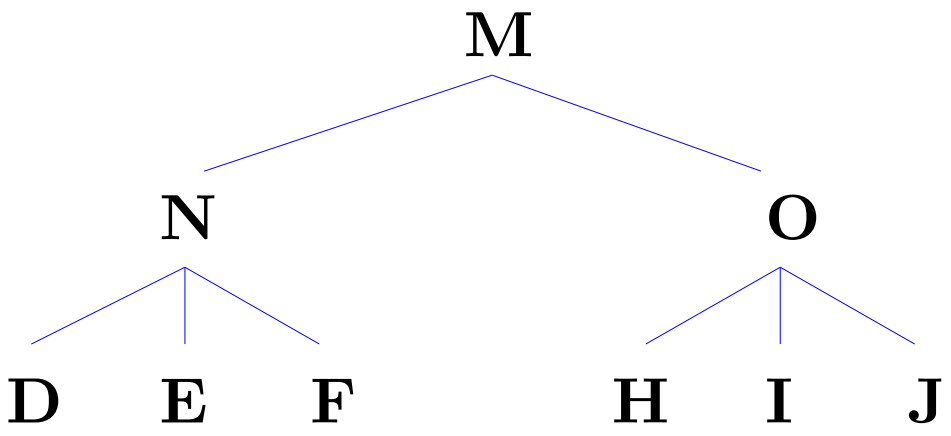
(English, German, Turkish, etc.)”

Tree geometry terminology



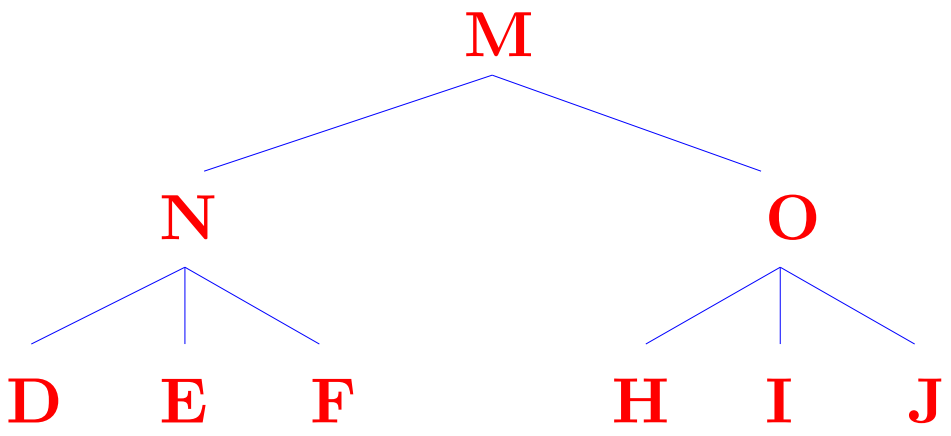
Branches

Tree geometry terminology



Branches

Tree geometry terminology

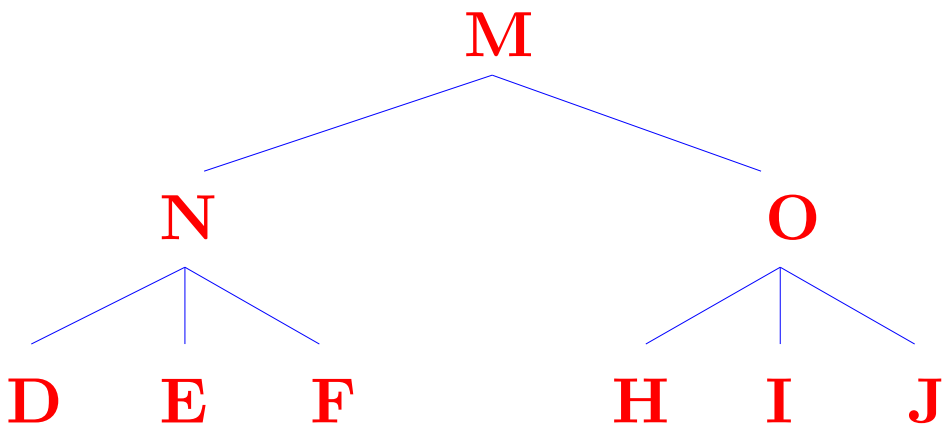


Branches

Nodes

Label

Tree geometry terminology

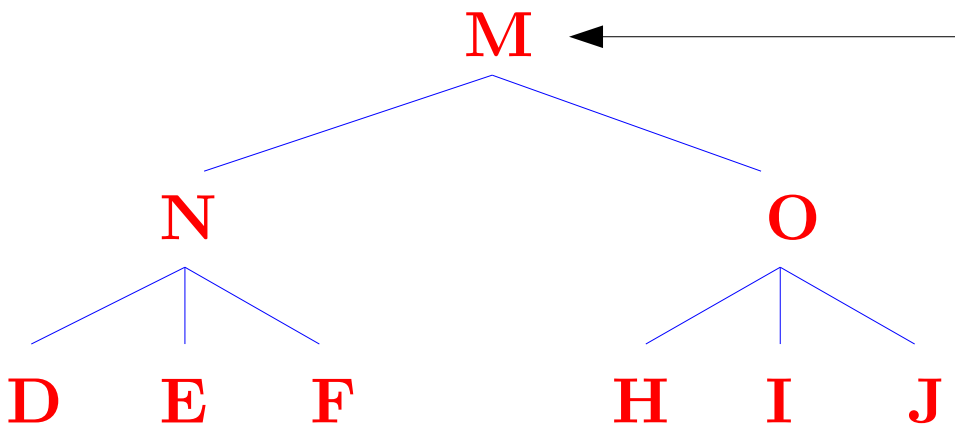


Branches

Nodes

Label = name given to a node

Tree geometry terminology

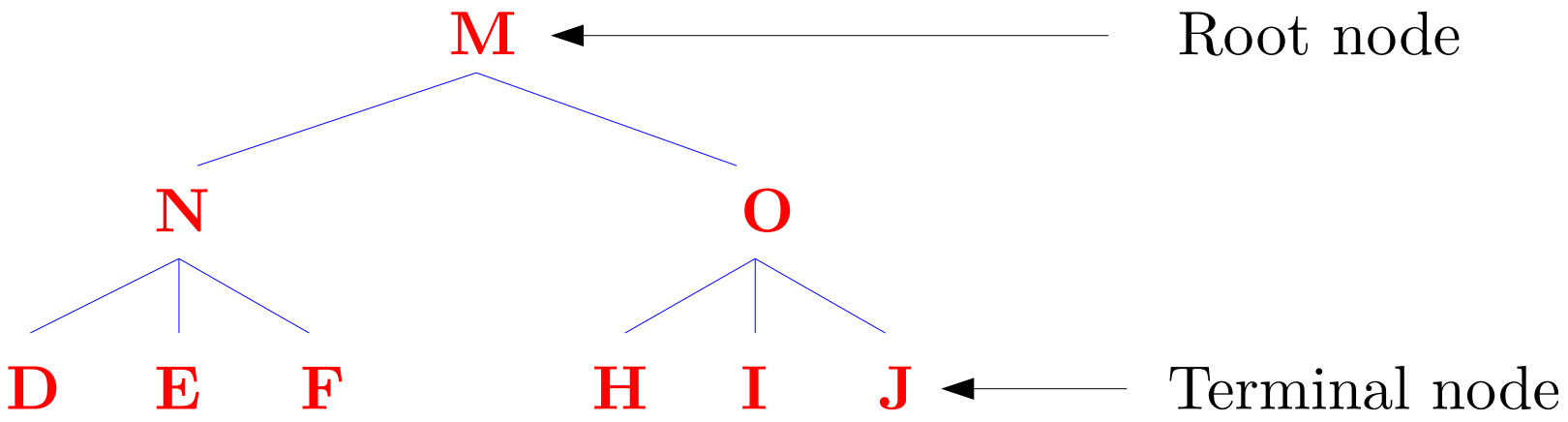


Branches

Nodes

Label = name given to a node

Tree geometry terminology

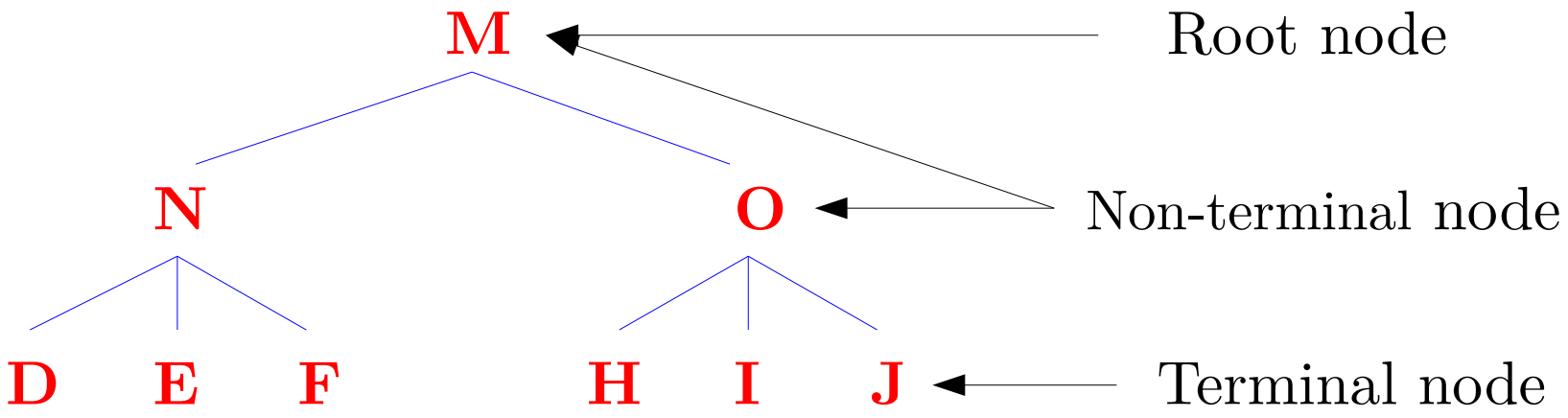


Branches

Nodes

Label = name given to a node

Tree geometry terminology

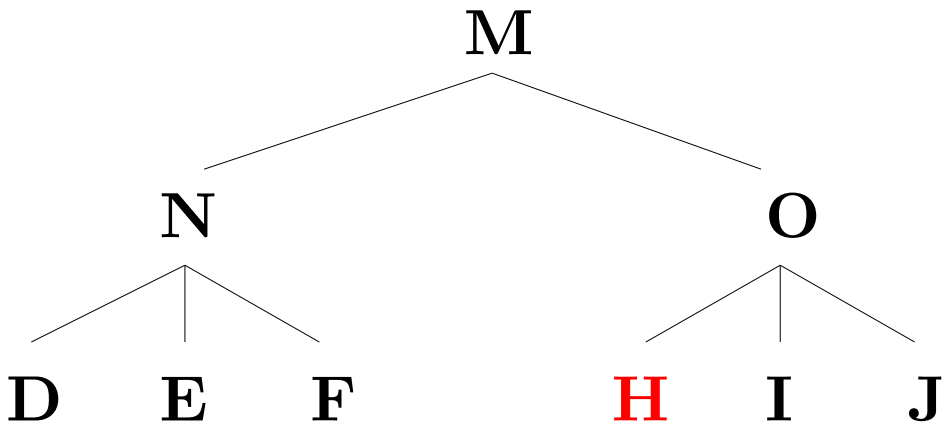


Branches

Nodes

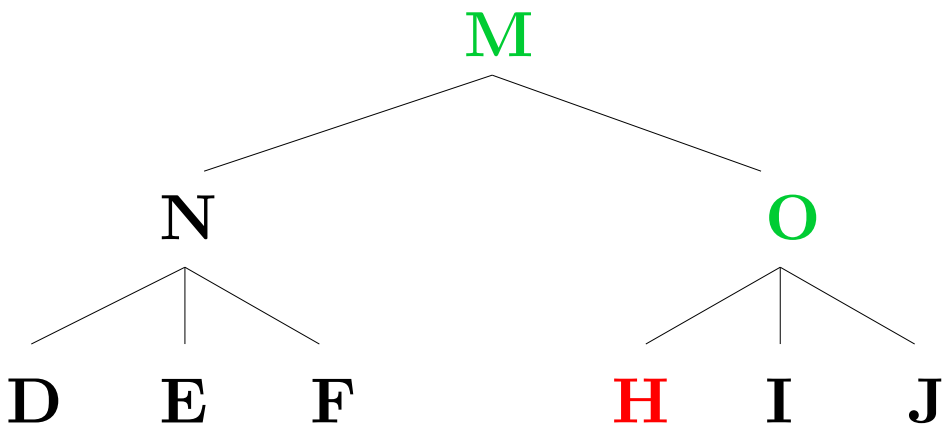
Label = name given to a node

Tree geometry terminology



Dominance

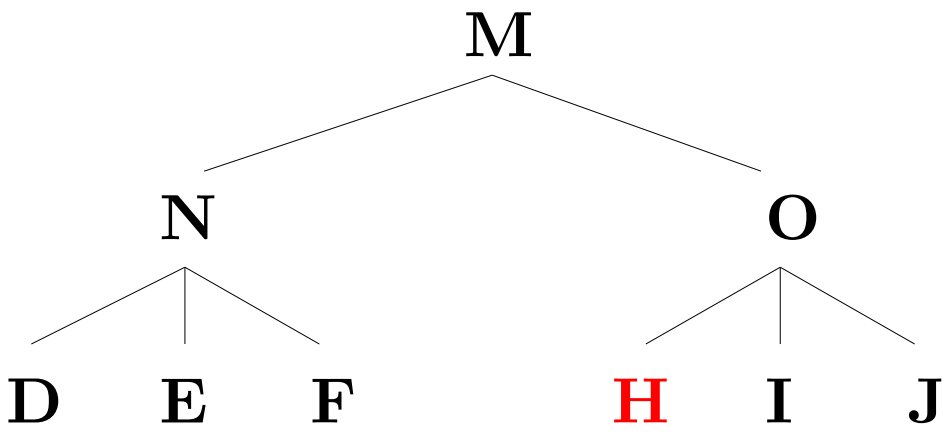
Tree geometry terminology



Dominance =

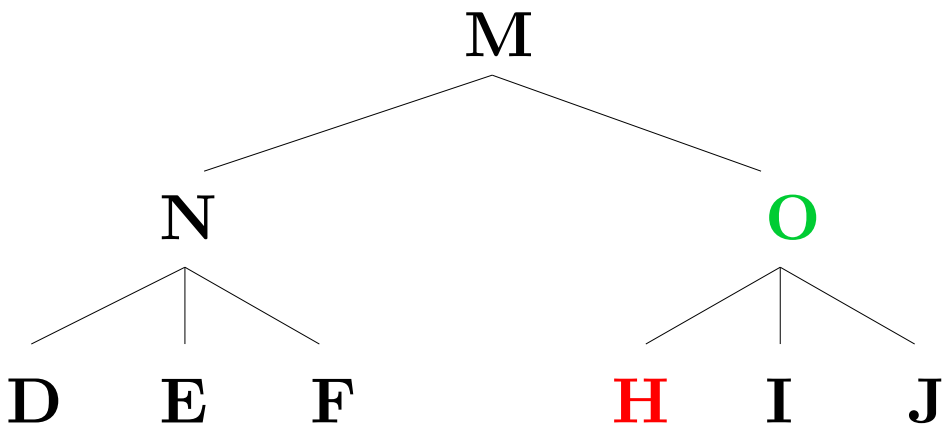
N1 dominates N2 if N1 is higher in the tree than N2 **and** there is a path connecting N1 to N2.

Tree geometry terminology



Immediate dominance =

Tree geometry terminology



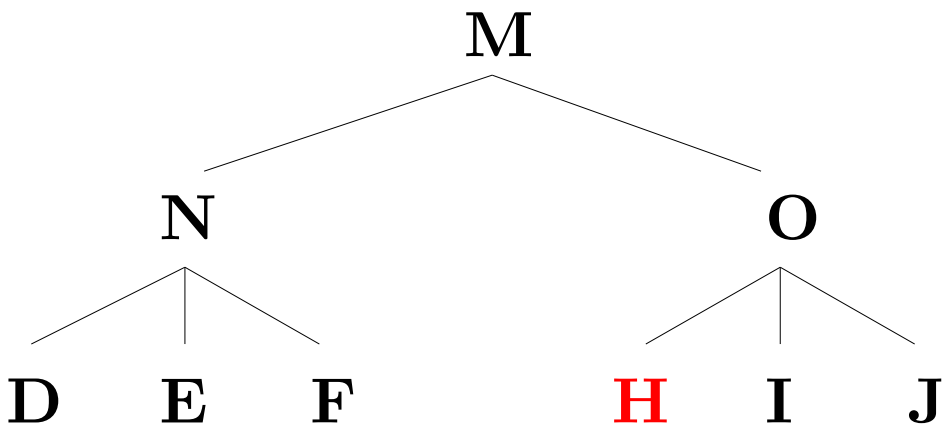
Immediate dominance =

N1 dominates N2 if N1 is higher in the tree than N2 **and**
there is a path connecting N1 to N2 **and**
there is no node on the path between N1 and N2

O is the mother of **H**

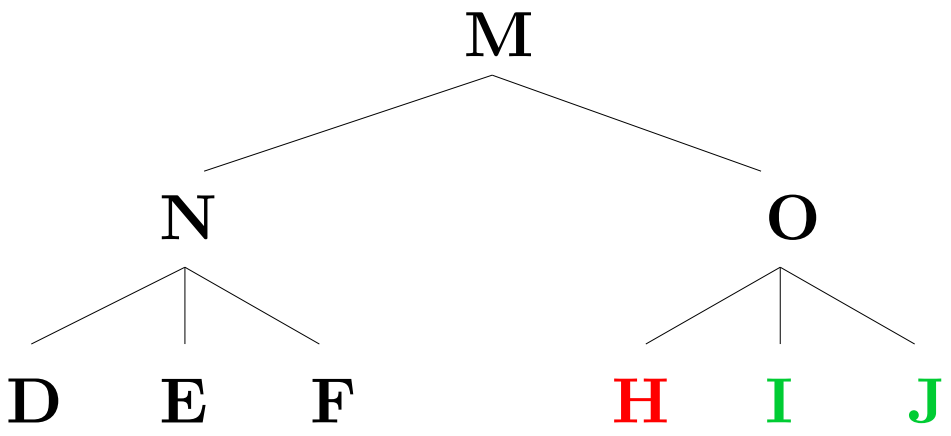
H is the daughter of **O**

Tree geometry terminology



Sister =

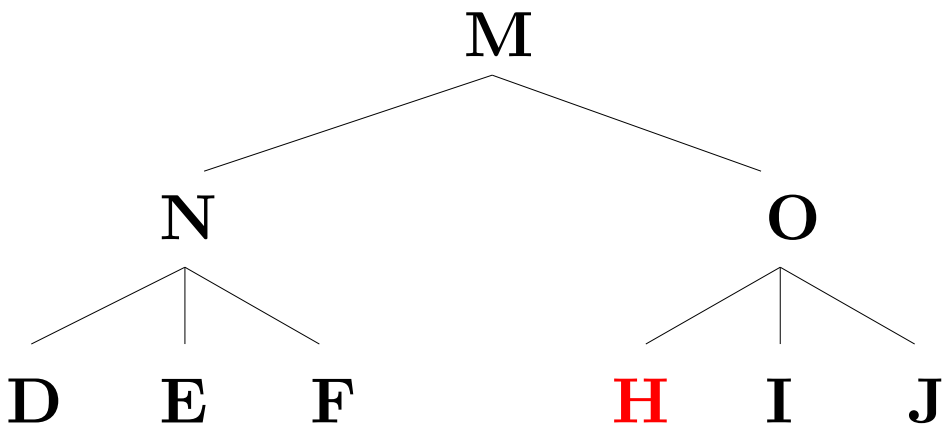
Tree geometry terminology



Sister =

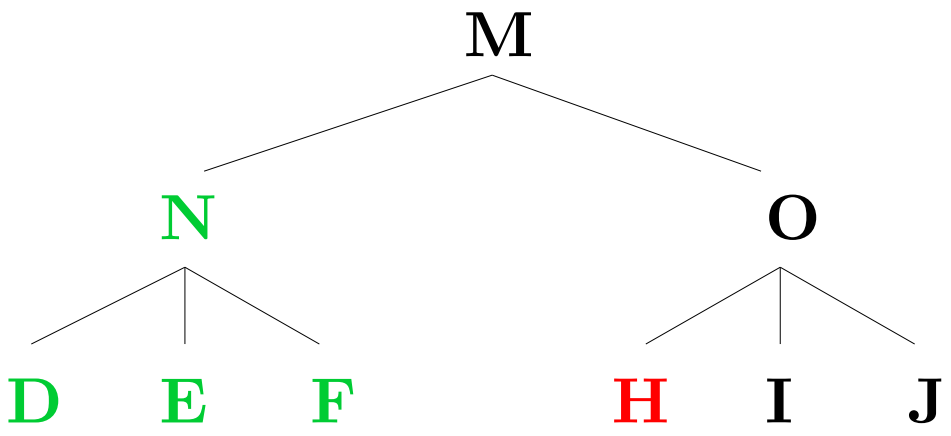
N1 and N2 are sisters if they have the same mother

Tree geometry terminology



Precedence =

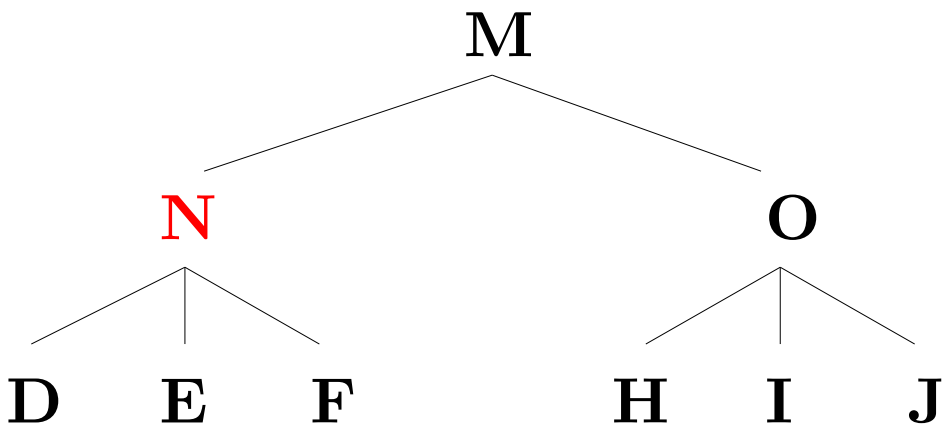
Tree geometry terminology



Precedence =

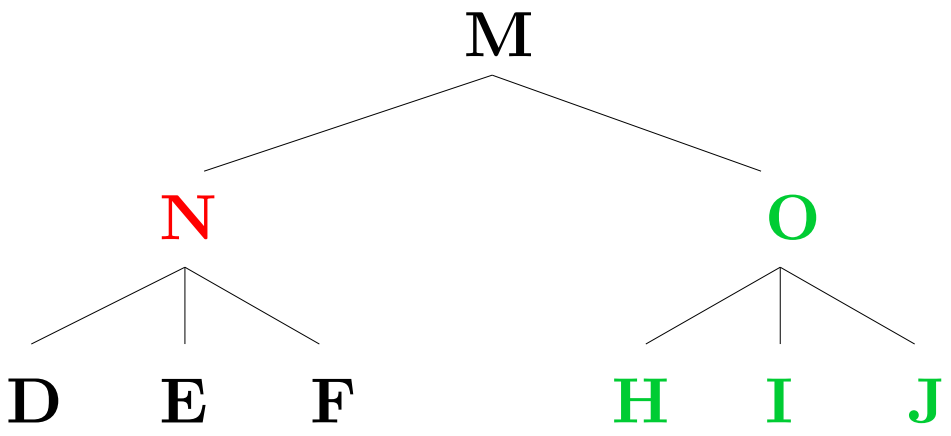
N1 precedes N2 if N1 does not dominate N2 and N1 is the leftward sister (or is dominated by the leftward sister) of N2 (or a node that dominates N2).

Tree geometry terminology



C-command =

Tree geometry terminology



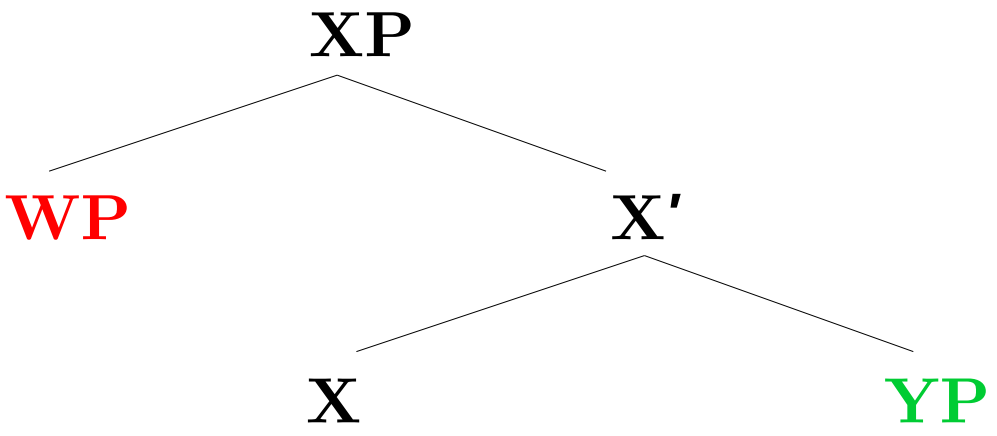
C-command =

If N1 has sisters, then it c-commands its sisters and all of their descendants

If N1 doesn't have any sisters, then it c-commands everything that its mother c-commands

Symmetric c-command = when N1 and N2 c-command each other

Tree geometry terminology



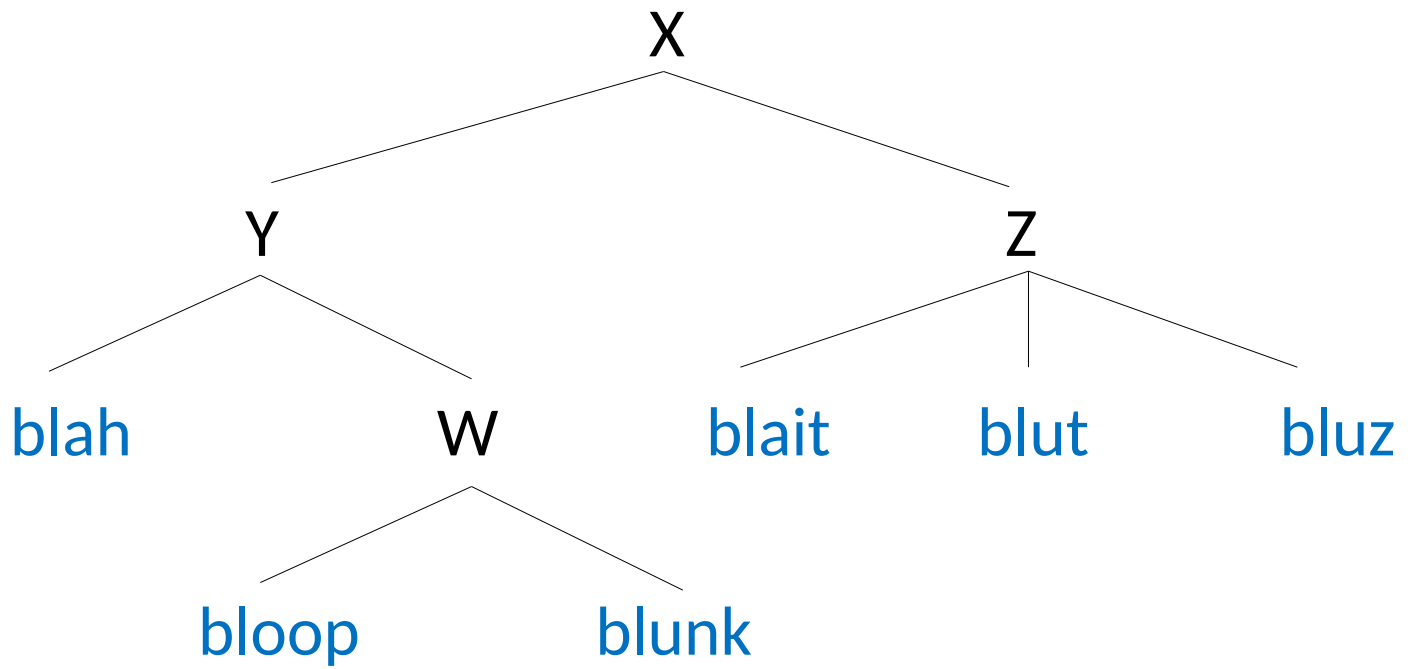
Complement =

Sister to X

Specifier =

Sister to X′

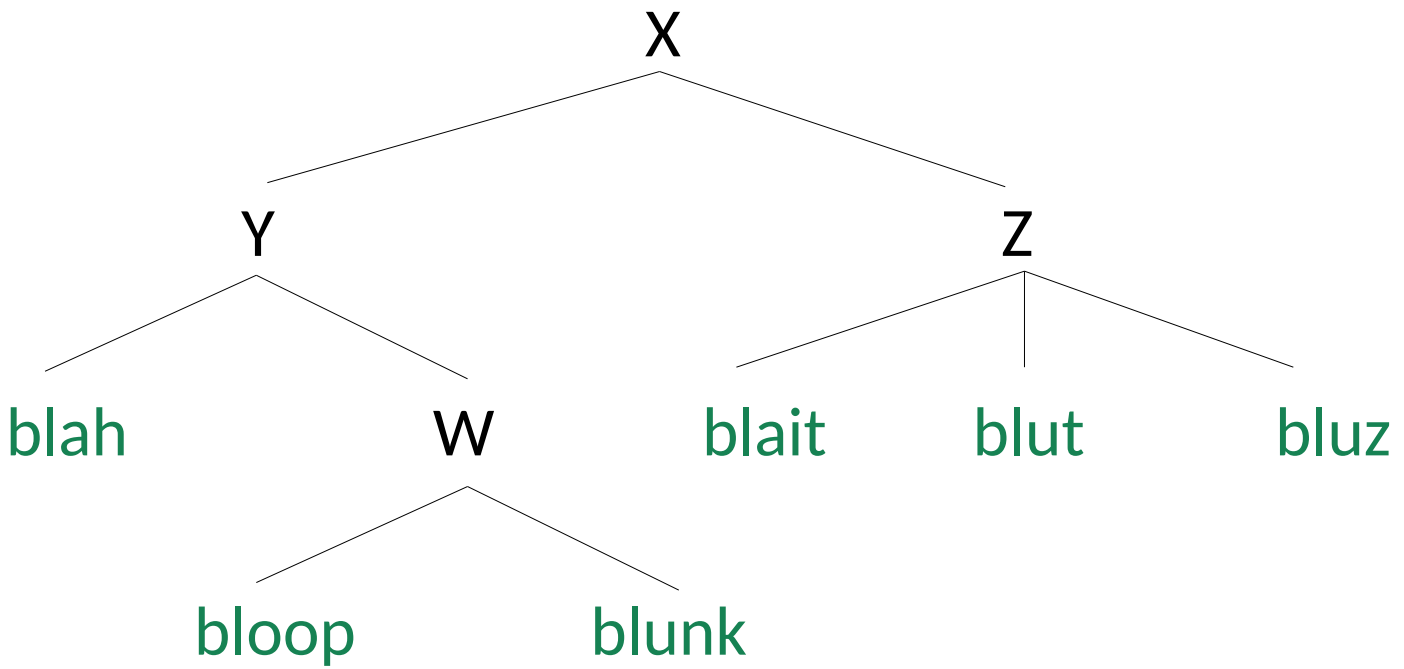
Tree geometry terminology



A constituent:

Any node **N** plus all of the nodes that **N** dominates (if **N** dominates something; it doesn't have to)

Tree geometry terminology

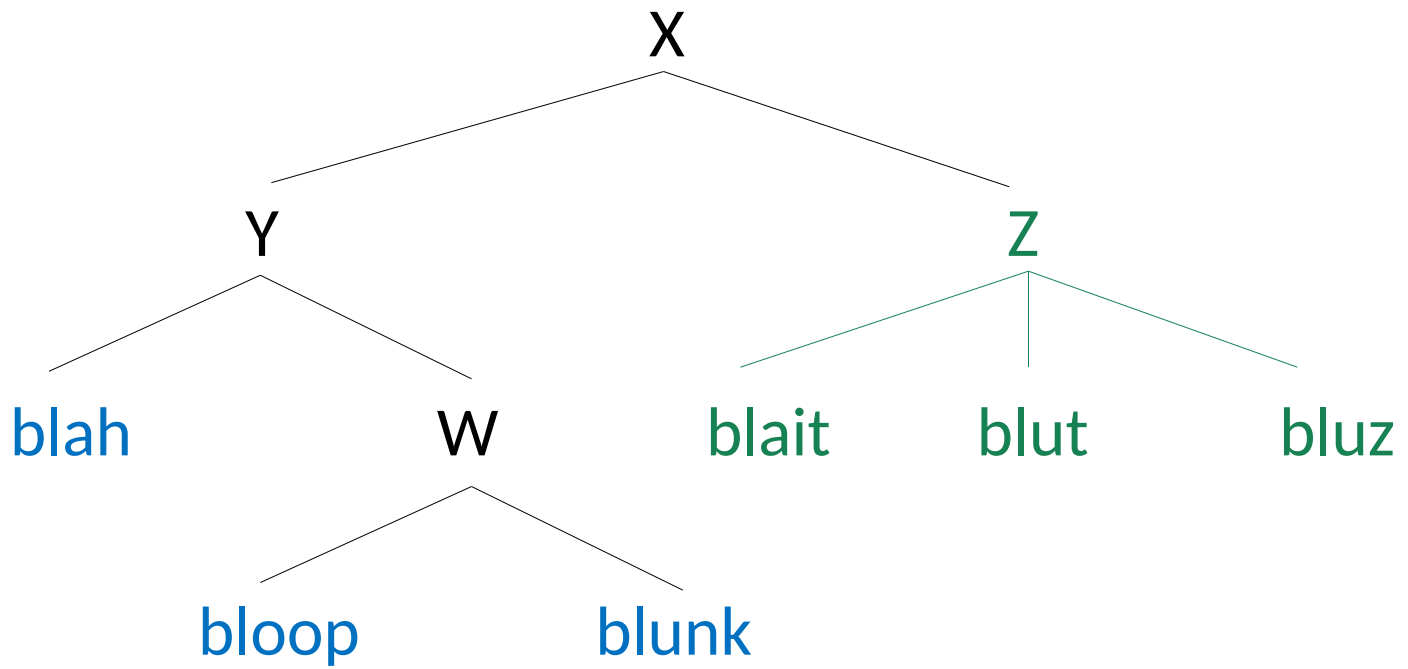


A constituent:

Any node **N** plus all of the nodes that **N** dominates (if **N** dominates something; it doesn't have to)

- All words are constituents (trivially so!)

Tree geometry terminology

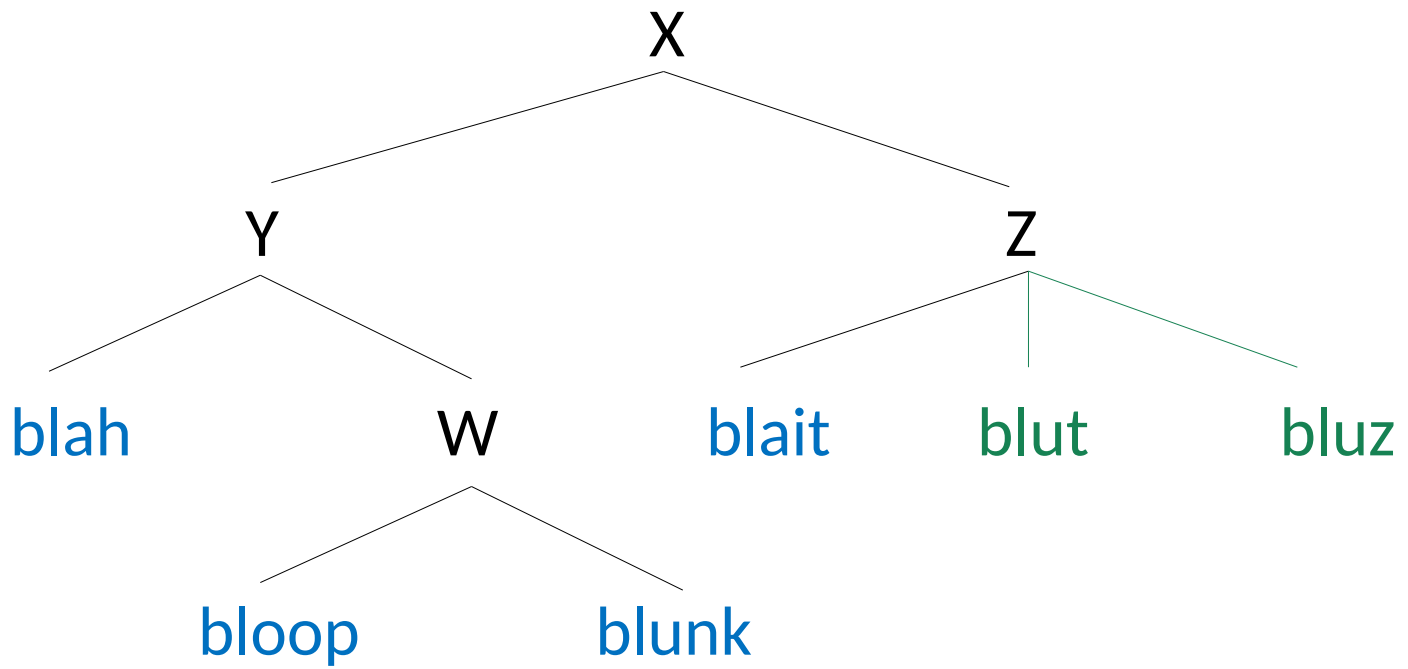


A constituent:

Any node **N** plus all of the nodes that **N** dominates (if **N** dominates something; it doesn't have to)

- The string of words blait blut bluz forms a constituent (represented by **Z**)

Tree geometry terminology



A constituent:

Any node **N** plus all of the nodes that **N** dominates (if **N** dominates something; it doesn't have to)

- The string blut bluz does **not** form a constituent

Dominance N1 dominates N2 if N1 is higher in the tree than N2 and there is a path connecting N1 to N2.

Mother N1 is the mother of N2 if N1 immediately dominates N2

Sister N1 and N2 are sisters if the have the same mother

Precedence N1 precedes N2 if N1 is leftward of N2

C-command If N1 has sisters, then N1 c-commands its sisters and their descendants

If N1 doesn't have sisters, then N1 c-commands everything its mother c-commands

Symmetric c-command N1 and N2 c-command each other

A constituent: Any node **N** plus all of the nodes that **N** dominates (if **N** dominates something)