5. Structural Relations

The Parts of a Tree, Domination, Exhaustive Domination, Immediate Domination, Precedence, C-command, Grammatical Relations

ENG467: Syntax and Structures of Language

Introduction

- In our earlier lectures, we developed the notion of constituency.
- Constituents are groups of words that function as single units.
- In order to systematically identify these, we proposed a set of rules.
- These rules generate trees, which in turn represent constituency.
- In these lectures, we will be representing sentences in a tree diagrams.

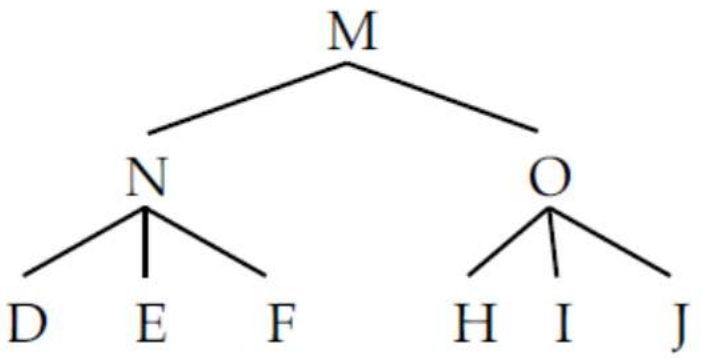
- There are actually two very good reasons why we should do this (draw tree diagrams).
- **First**, by considering the geometry of trees, we can assign names to the various parts and describe how the parts relate to one another.
  - For example, in the last lectures we were only able to give a vague definition of the term *constituent*.
  - In these lectures, we'll be able to give a precise description.
- **Second**, it turns out that there are many syntactic phenomena that can be understood through tree diagrams.
  - One of the most obvious of these refers to anaphors.
  - Anaphors can only appear in certain positions in the tree.

However, the distribution of anaphors and other types of nouns will be discussed in other modules.

1. THE PARTS OF A TREE

Let's start with a very abstract tree drawing:

1)



This tree would be generated by the rules in (2):

2) M $\longrightarrow$ N O

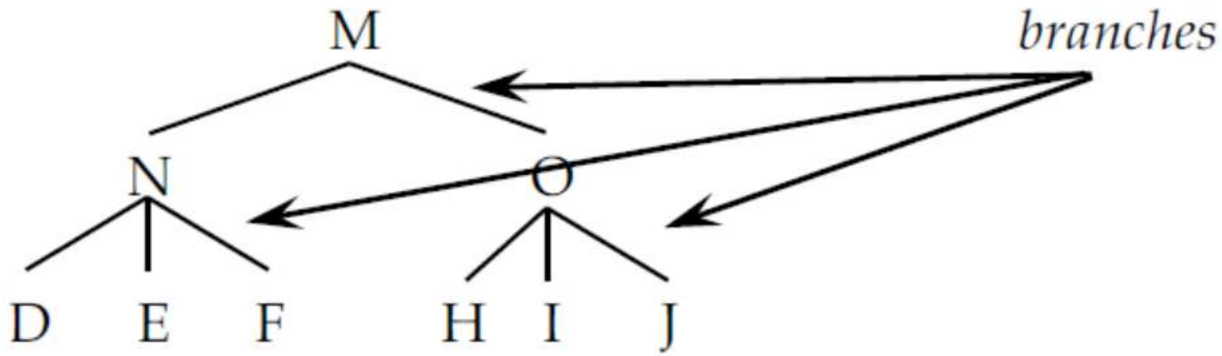
N $\longrightarrow$ D E F

O $\longrightarrow$ H I J

- Now we can describe the various parts of this tree.
- The lines in the tree are called ***branches***.
- A formal definition of branch is given in (3), and the branches are marked in (4):

3) ***Branch:*** A line connecting two parts of a tree.

4)



- The end of any branch is called a ***node***.
- Both ends are called nodes.
  - For example, N and F are both called nodes of a branch.

- Any time two or more branches come together, this is also called a node:

5) ***Node*****:** The end of a branch.

- A node that has branches (daughters) below it is said to be **branching (Non-terminal)**;
- A node that has **no** branch (daughter) below it is said to be **non-branching (Terminal)**.

- Nodes in a tree are labeled.
- In the tree above, M, N, O, D, E, F, H, I, J are the ***labels*** for the nodes that make up the tree. This is very abstract of course.
- In the last module, we looked at the various parts of speech (N, V, A, P, etc.) and the phrasal categories associated with them (NP, VP, AP, PP, etc.).
- These are the labels in a real syntactic tree.

6) ***Label*****:** The name given to a node.

- There are actually different kinds of nodes that we'll want to make reference to.
- The first of these is called the ***root node***.
    - The root node doesn't have any branch on top of it.
    - There is only ever one root node in a sentence.

- In the trees we looked at in the last module, the root node was almost always the TP (sentence) node.

7) ***Root node*** *(preliminary)*: The node with no line on top of it.

- At the opposite end of the tree are the nodes that don't have any lines underneath them. If the tree analogy were to really hold up, we should call these "leaves." More commonly, however, these are called ***terminal nodes***.

  8) ***Terminal node*** *(preliminary):* Any node with no branch underneath it.

- Any node that isn't a terminal node is called a ***non-terminal node***:

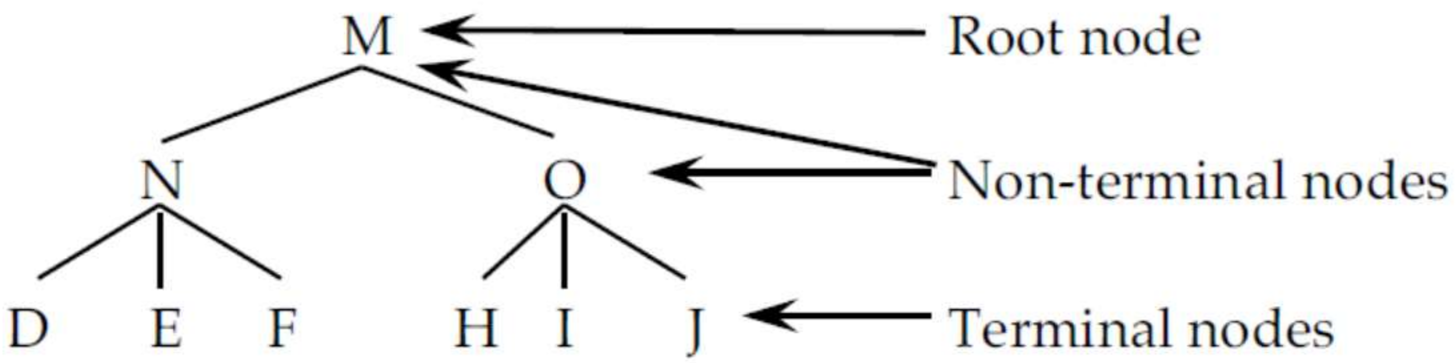
  9) ***Non-terminal node*** *(preliminary)*: Any node with a branch underneath it.

- Notice that the root node is also a non-terminal node by this definition.

- After we add some definitions in the next module, we'll have reason to reformulate the definitions of root, terminal and non-terminal nodes, but for now these should give you the basic idea.

- In (10), we have a tree where the root node, the terminal nodes, and the non-terminal nodes are all marked.

10)



- In this tree, M is the **root node**.

  M, N, and O are **non-terminals,** and

  D, E, F, H, I, and J are **terminal nodes**.

Let's have a glance at the Tree Terminologies of a tree structure

Branch: A line connecting two nodes

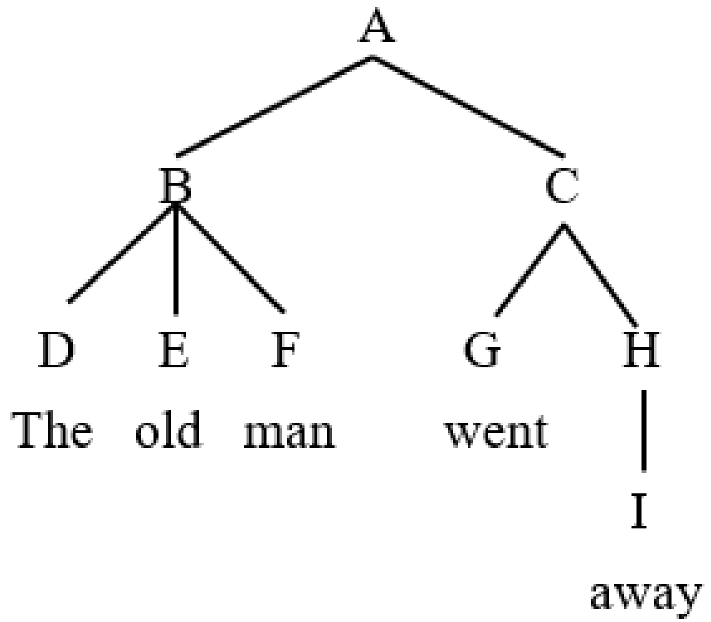
Node: The end of a branch – ABCDEFGHI

Label: Names for the nodes

Root node

Non-terminal (branching) nodes

Terminal (non-branching) nodes



Root node : A node that dominates any other nodes.

Non-terminal (branching) nodes : A node that is dominated by another node and also dominates other nodes

Terminal (non-branching) nodes : A node that does not dominate any other node.

- We now have **all the terms we need to describe** the **various parts of a tree**.
- The lines are called **branches**.
- The ends of the lines are called **nodes**, and each of the nodes has a **label**.
- Depending upon where the node is in the tree, it can be a root node (the top), a terminal (the bottom), or a non-terminal (any node except the bottom).

- Next we turn to *a set of terms and descriptions that will allow us to describe the relations that hold between these parts*. Because we are talking about a tree structure here, these relations are often called ***structural relations***.

Here

2. DOMINATION

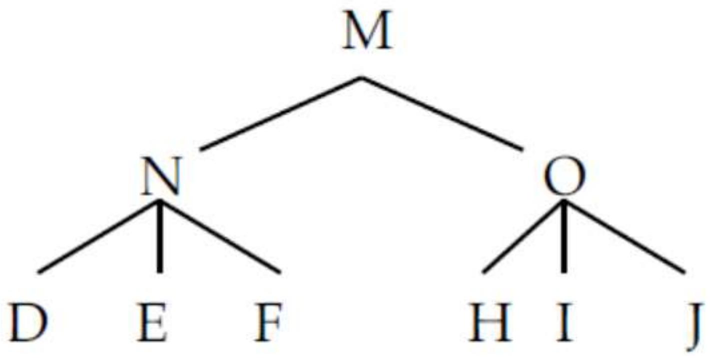
2.1 Domination

- Some nodes are higher in the tree than others.
- This reflects the fact that trees show a hierarchy of constituents.
- In particular, we want to talk about nodes that are higher than one another *and* are connected by a branch.
- The relation that describes two nodes that stand in this configuration is called ***domination***.
- A node that sits atop another and is connected to it by a branch is said to dominate that node.

11) ***Domination*:** Node A dominates node B if and only if A is higher up in the tree than B and if you can trace a line from A to B going only downwards.

- In (12), M dominates all the other nodes (N, O, D, E, F, H, I, J).
- N dominates D, E, and F, and O dominates H, I, and J.
- O does not dominate F, as you can see by virtue of the fact that there is no branch connecting them.

12)



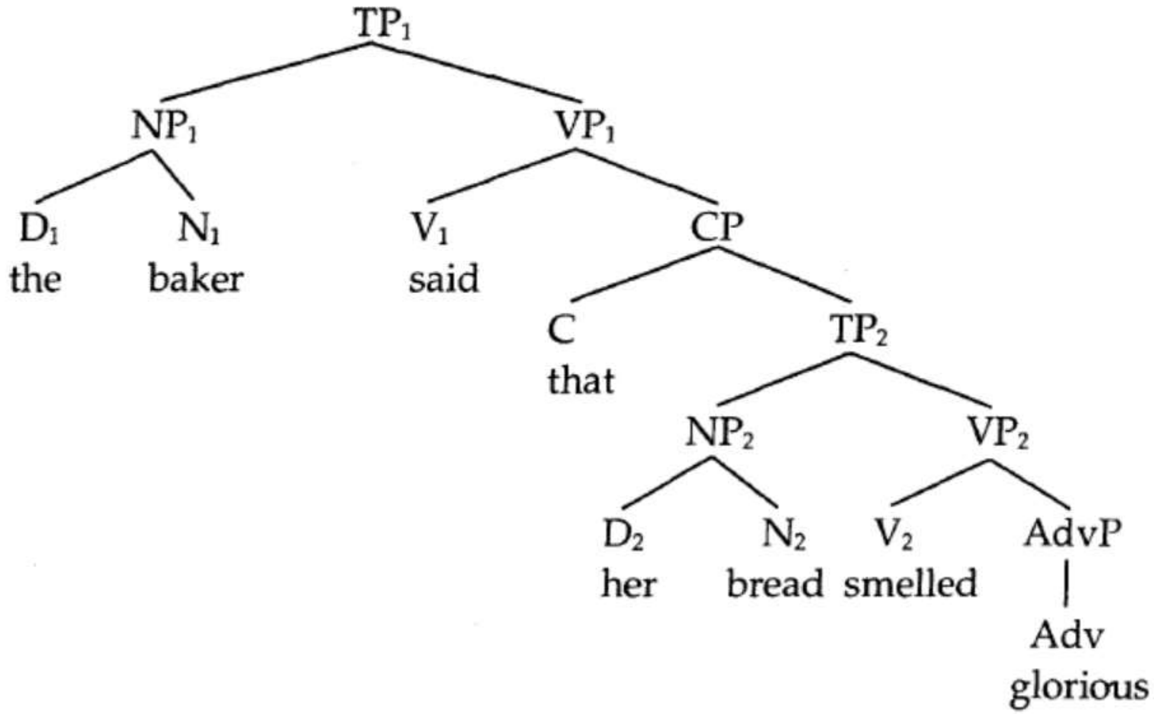
- Domination is essentially a containment relation. The phrasal category N contains the terminal nodes D, E, and F. Containment is seen more clearly when the tree is converted into a bracketed diagram:

13) [M [N D E F] [O H I J]]

- In (13) the brackets associated with N ([N D E F]) contain the nodes D, E, and F.
- The same holds true for O which contains H, I, and J.
- M contains both N and O and all the nodes that they contain.
- So domination is a technical way of expressing *which categories belong to larger categories.*

Dominance exercise:

1. Study the following tree carefully and then answer the questions about it that follow:



1) List all the nodes that dominate each of the following items:

a) D_1 *the* b) D_2 *her* c) N_1 *baker* d) N_2 *bread* e) V_1 *said* f) V_2 *smelled* g) Adv *glorious*

h) C *that* i) TP_1 (if there are any) j) TP_2 k) NP_1 l) NP_2 m) VP_1

n) VP_2 o) CP p) AdvP

2) What is the root node?

3) List all the terminal nodes.

4) List all the non-terminal nodes.

5) List all the nodes that the following nodes dominate:

a) VP2 b) CP c) NP1

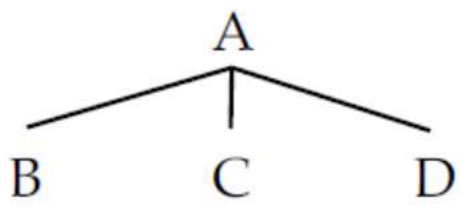
2.2 Exhaustive Domination

- In the last module, we developed an intuitive notion of constituent.
- The relation of domination actually allows us to be a little more rigorous and develop a formal notion of constituency. In order to do this, we need another definition, ***exhaustive domination*:**

14) ***Exhaustive domination*:** Node A exhaustively dominates a *set* of terminal nodes {B, C, ..., D}, provided it dominates all the members of the set (so that there is no member of the set that is not dominated by A) *and* there is no terminal node G dominated by A that is not a member of the set.

- This is a rather laborious definition. Let's tease it apart by considering an example.

15)



- What we are concerned with here is a *set* of nodes and whether or not a given node dominates the entire set.

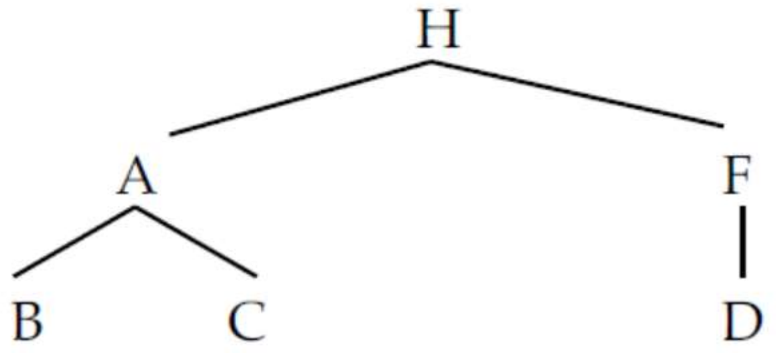
- Sets are indicated with curly brackets {}.
- Start with the set of terminal {B, C, D}.
- In (15) all members of the set {B, C, D} are dominated by A;

there is no member of the set that isn't dominated by A.

This satisfies the first part of the definition in (14).

- Turning to the second part, A *only* dominates these terminal nodes and no other terminals.
  There is no node G dominated by A that is not a member of the set.
- This being the case we can say of the tree in (14) that A exhaustively dominates the set {B, C, D}.

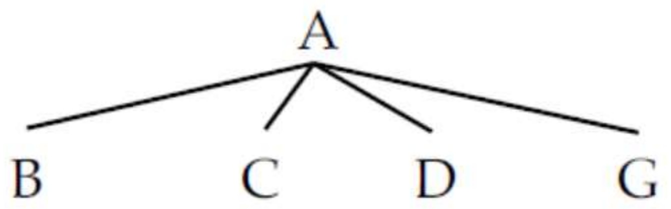
- Let's turn to a different tree now.
  16)



- Again let's consider whether A exhaustively dominates the set {B, C, D}.
- In (16), one member of the set, D, is not dominated by A.
- Thus the set {B, C, D} is *not* exhaustively dominated by A.

- The reverse situation is seen in (17):

17)



- While it is the case that in (17), B, C, and D are all immediately dominated by A, there is also the node G, which is not a member of the set {B, C, D}, so the set {B, C, D} is not exhaustively dominated by A (although the set {B, C, D, G} is).
- On a more intuitive level, *exhaustive domination holds between a set of nodes and their mother*. Only when the entire set (and only that set) are immediately dominated by their mother can we say that the mother exhaustively dominates the set.

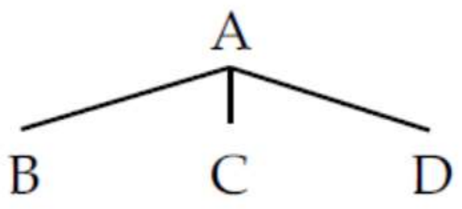
- Look carefully at the structures in (15), (16), and (17).
- In (15) you'll see that the set {B, C, D} forms a constituent (labeled A).
- In (17), that set does not form a constituent (although the set is part of a larger constituent in that tree).
- In (17), there is no sense in which B, C, and D form a unit that excludes G.
- It seems then that the notion of constituency is closely related to the relation of exhaustive domination. This is reflected in the following formal definition of a constituent:

18) ***Constituent*:** A set of terminal nodes exhaustively dominated by a particular node.

- If we look at the tree in (16) again, you can see that each constituent meets this definition. The set of nodes exhaustively dominated by A is {B, C}, which is the set of terminals that make up the A constituent.
- Similarly, the constituent F is made up of the set {D}, which is exhaustively dominated by F;
- finally, H exhaustively dominates {B, C, D} (remember the definition is defined over *terminals*, so A and F don’t count) which is the constituent that H represents.

- Before turning to some other structural relations, it is important to look at one confusing piece of terminology.
- This is the distinction between ***constituent*** and ***constituent of*.**
- A ***constituent***, as defined in (18), is a set of nodes exhaustively dominated by a single node.
- A ***constituent of***, by contrast, is a *member* of the constituent set. Consider the tree in (19):

19)

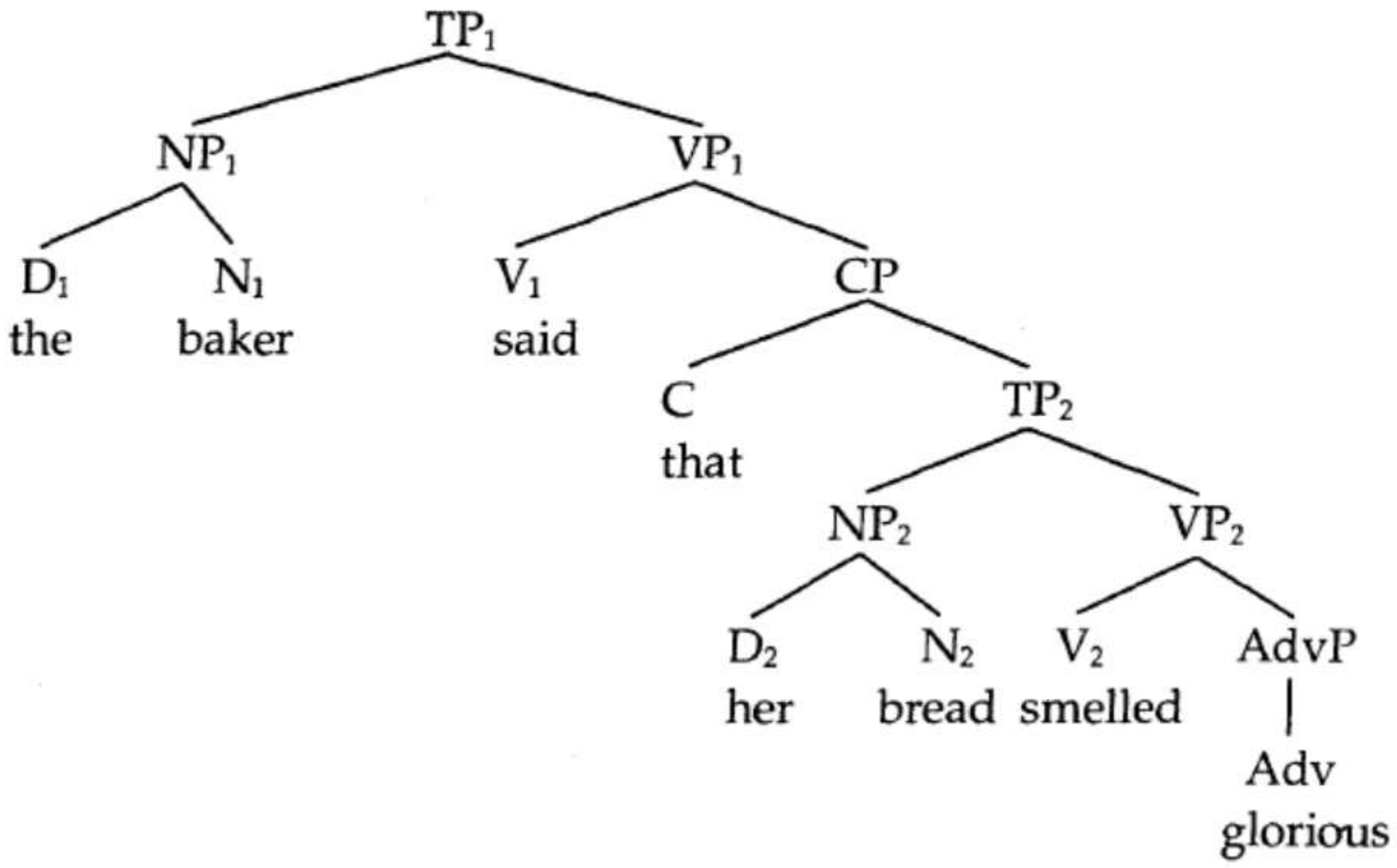


- Here we have the constituent A, which exhaustively dominates the set {B, C, D}.
- Each member of this set is called a "constituent of A." So B is a constituent of A. "Constituent of" boils down to domination. A dominates B; therefore B is a constituent of A:

20) ***Constituent of*:** B is a constituent of A if and only if A dominates B.

Exhaustive Dominance exercise:

Q. Study the tree carefully and answer the following questions.



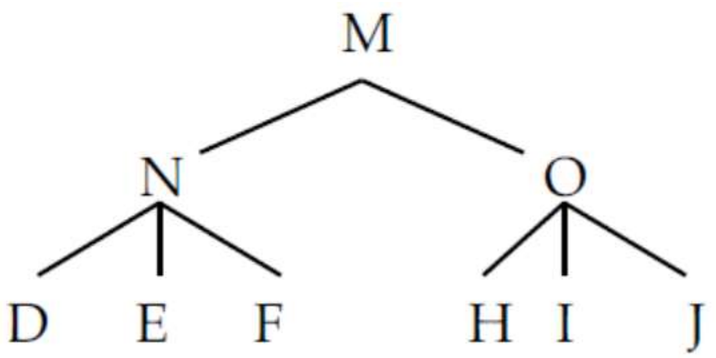
Possible questions from the tree:

1) In the tree, is the set of terminals {N_1, N_2} exhaustively dominated by a single node? If so, which one?
2) In the tree, is the set {D_1, N_1} exhaustively dominated by a single node? Which one?
3) In the tree, is the set {V_2, Adv} exhaustively dominated by a single node? Which one?
4) In the tree, is the set {D_2, N_2, V_2, Adv} exhaustively dominated by a single node? Which one?
5) In the tree, is the set {D_1, N_1; V_I} exhaustively dominated by a single node? Which one?
6) In the tree, is the set {D_1} exhaustively dominated by a single node? Which one?
7) In the tree, is the set {C, D_2, N_2, V_2, Adv} exhaustively dominated by a single node? Which one?
8) What is the set of terminal nodes exhaustively dominated by VP_1?
9) Is the string *that her bread* a constituent? Explain your answer using the terminology of exhaustive dominance.
11) Is NP_1 a constituent of TP_1?
12) Is NP_2 a constituent of TP_1?
13) Is NP_1 a constituent of TP_2?
14) Is NP_2 a constituent of TP_2?
15) Is V_2 a constituent of CP?
16) Is VP_2 a constituent of CP?
17) Are both Adv and AdvP constituents of VP_2?

2.3 Immediate Domination

- Domination is actually quite a general notion:
- In (21), M dominates all of the nodes under it.

21)



- In certain circumstances we might want to talk about relationships that are smaller and more local.
- This is the relationship of *immediate dominance*.
- A node immediately dominates another if there is only one branch between them.

22) ***Immediately dominate:*** Node A immediately dominates node B if there is no intervening node G that is dominated by A, but dominates B. (In other words, A is the first node that dominates B.)

- In (21), M dominates all the other nodes in the tree, but it only immediately dominates N and O.
- It does not immediately dominate any of the other nodes because N and O intervene.

- There is an informal set of terms that we frequently use to refer to immediate dominance.
- This set of terms is based on the fact that syntactic trees look a bit like family trees.
- If one node immediately dominates another, it is said to be the ***mother;*** the node that is immediately dominated is called the ***daughter***.

- In the tree above in (21), N is D's mother and D is N's daughter. We can even extend the analogy (although this is pushing things a bit) and call M as D's grandmother.

23) ***Mother**:* A is the mother of B if A immediately dominates B.

24) ***Daughter**:* B is the daughter of A if B is immediately dominated by A.

Closely related to these definitions is the definition of *sister:*

25) ***Sisters**:* Two nodes that share the same mother.

- With this set of terms in place we can now redefine our definitions of root nodes, terminal nodes, and non-terminals a little more rigorously:

26) ***Root node*** *(revised):* The node that dominates everything but is dominated by nothing.
(The node that is no node's daughter.)

27) ***Terminal node*** *(revised):* A node that dominates nothing.
(A node that is not a mother.)

28) ***Non-terminal node*** *(revised):* A node that dominates something and also dominated by other node.
(A node that is a mother.)

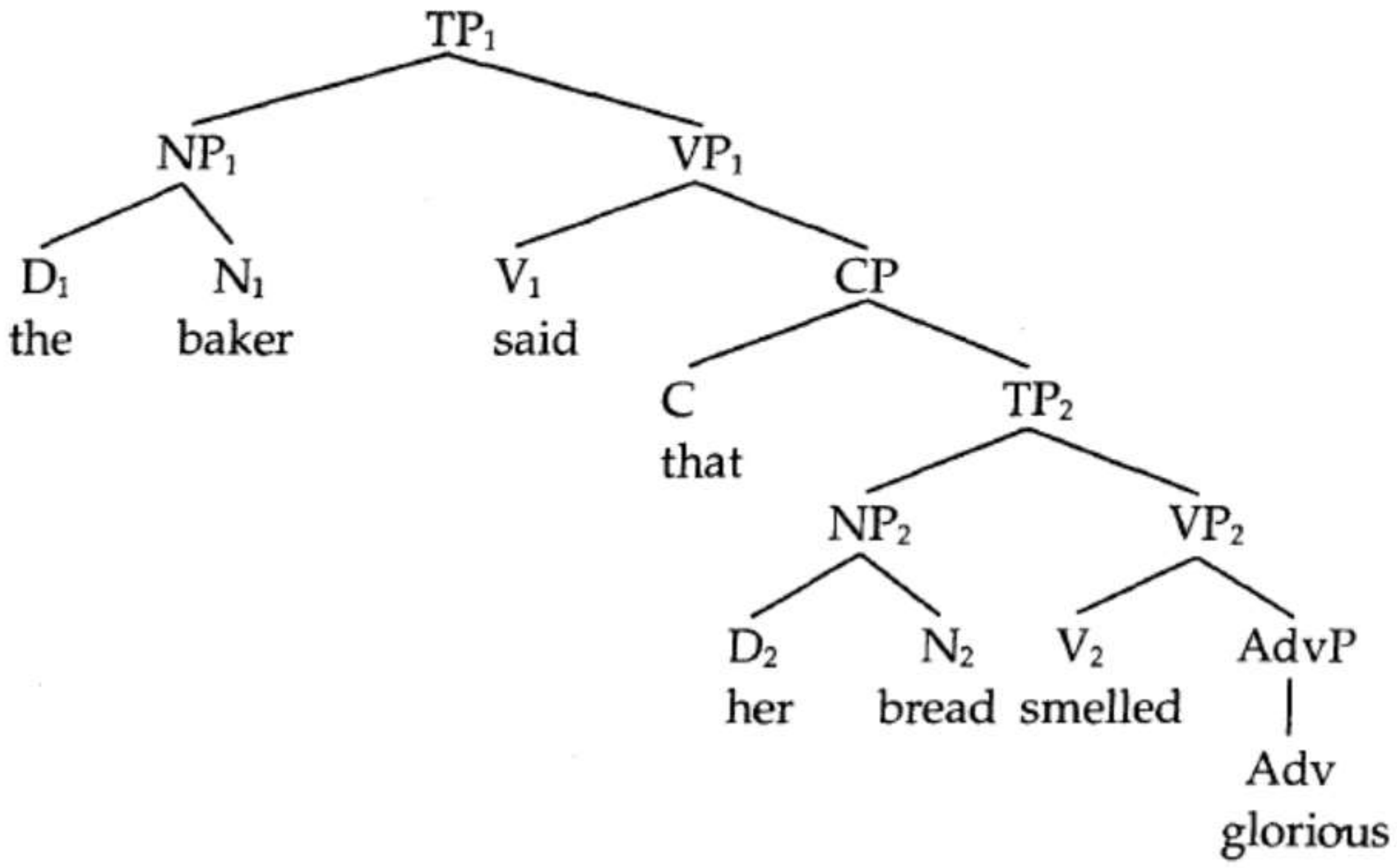
- We defined "constituent" in terms of dominance, and from that we derived the "constituent of" relation (essentially the opposite of dominance).
- We can also define a local variety of the "constituent of" relation that is the opposite of immediate dominance:

29) *Immediate constituent of:* B is an immediate constituent of A if and only if A immediately dominates B.

- This ends our discussion of the vertical axis of syntactic trees. Next, we consider horizontal relations.

Immediate Dominance exercise:

Q. Study the tree carefully and answer the following questions.



Q. What immediately dominate what?

3. PRECEDENCE

- Syntactic trees don't only encode the hierarchical organization of sentences, they also encode the linear order of the constituents.
- Linear order refers to the order in which words are spoken or written (left to right if you are writing in English). Consider the following rule:

30) M $\longrightarrow$ A B

- This rule not only says that M dominates A and B and is composed of A and B.
- It also says that A must precede B in linear order.
- A must be said before B, because it appears to the left of B in the rule. The relation of "what is said first" is called *precedence*.
- In order to define this rigorously we have to first appeal to a notion known as *sister precedence:*

31) ***Sister precedence:*** Node A sister-precedes node B if and only if both are immediately dominated by the same node, and A appears to the left of B.

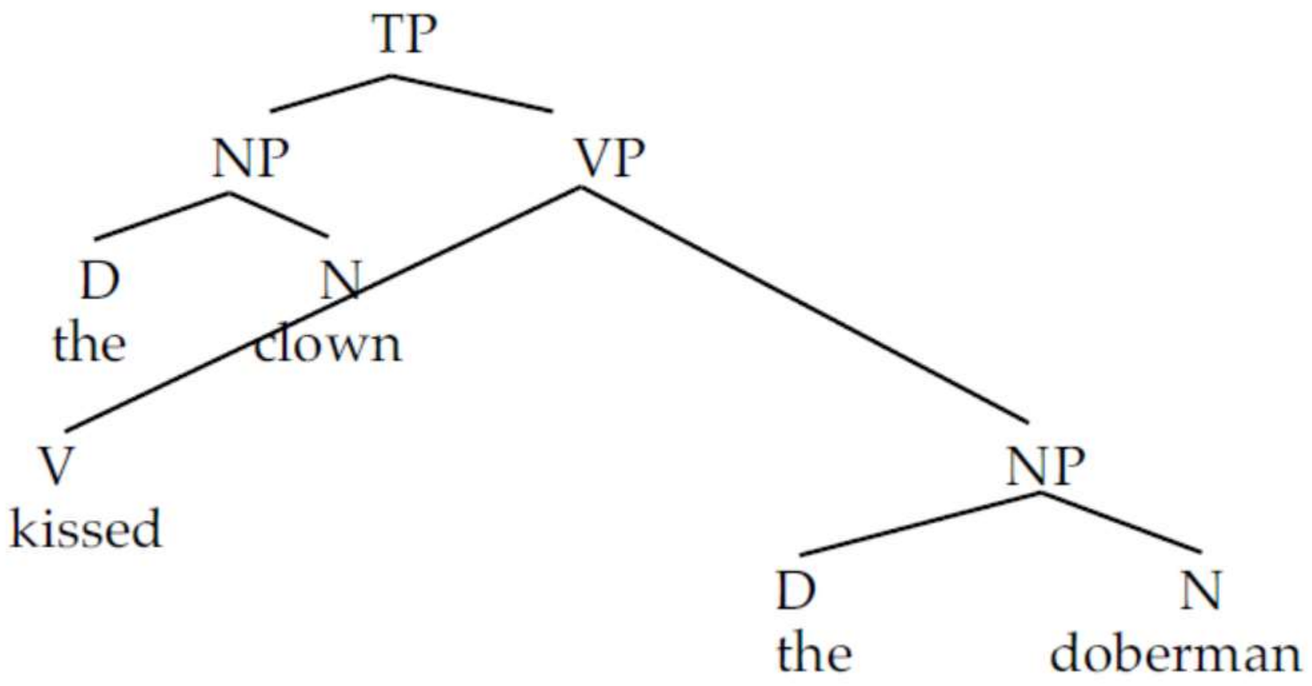
- The ordering in this definition follows from the order of elements within a phrase structure rule.
- If A is to the left of B in a phrase structure rule M $\longrightarrow$ A B, then A and B are immediately dominated by M, and are in the relevant order by virtue of the ordering within that rule.
- With this basic definition in mind we can define the more general precedence relation:

32) ***Precedence:*** Node A precedes node B if and only if neither A dominates B nor B dominates A *and* A (or some node dominating A) sister-precedes B or (some node dominating B).

- This definition is pretty complex, so let's break it apart.
- The **first bit of the definition** says **"neither A dominates B nor B dominates A."**
- The reason for this should be obvious on an intuitive level.
- Remember, dominance is a containment relation. If A contains B, there is no obvious way in which A could be to the left of B.
- Think of it this way. If you have a box, and the box has a ball in it, you can't say that the box is to the left of the ball. That is physically impossible. The box surrounds the ball. The same holds true for dominance. You can't both dominate and precede/follow.

- The **second part of the definition** says "A or some node dominating A sister-precedes B or some node dominating B."
- This may seem like an overly complex way to say "to the left," but there is a good reason we phrase it like this. This has to do with the fact that the terminals of a tree don't float out in space.
- Rather they are dominated by other nodes that might precede or follow themselves and other nodes. Consider the following tree drawn by a sloppy tree-drawer:

33)



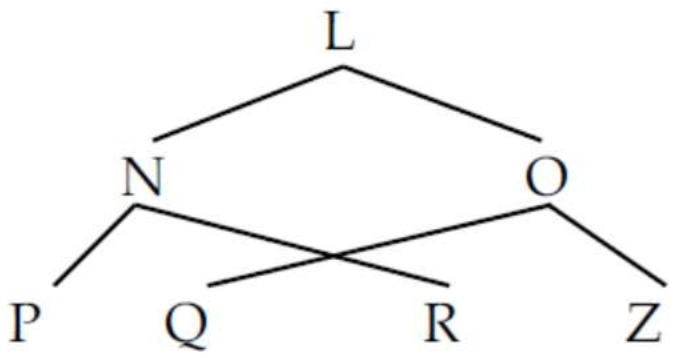
- In this sloppily drawn tree, the verb *kissed* actually appears to the *left* of the noun *clown*.
- However, we wouldn't want to say that *kissed* precedes *clown*; this is clearly wrong.
- The sentence is said "The clown kissed the doberman," where *kissed* follows *clown*.
- We guarantee this ordering by making reference to the material that dominates the nodes we are looking at. Let A = *clown* and B = *kissed*. Let's substitute those into the definition:

34) *[$_N$ clown]* or some node dominating *[$_N$ clown]* (in this case NP) sister-precedes *[$_v$ kissed]* or some node dominating *[$_v$ kissed]* (in this case VP).

This means that *[$_N$ clown]* precedes *[$_v$ kissed]*, because NP precedes VP.
Note that precedence holds over *all* nodes, not just terminals.
So [$_N$ *clown]* also precedes [$_{NP}$ *the doberman]*.

- The second clause of the definition also allows us to explain an important restriction on syntactic trees: ***You cannot allow branches to cross***.
- Trees like (35) are completely unacceptable (they are also impossible to generate with phrase structure rules- try to write one and you'll see).

35)



- In this tree, Q is written to the left of R, apparently preceding R, but by the definition of precedence given above, this tree is ruled out. Q is to the left of R, but *O which dominates Q* is not. In other words, you can't cross branches. Another way of phrasing this is given in (36):

36) ***No crossing branches constraint:*** If one node X precedes another node Y, then X and all nodes dominated by X must precede Y and all nodes dominated by Y.

- Just as in the dominance relation, where there is the special local definition called "immediate dominance," there is a special local form of precedence called *immediate precedence:*

  37) *Immediate precedence:* A immediately precedes B if there is no node G that follows A but precedes B.

- Consider the string given in (38) (assume that the nodes dominating this string meet all the criteria set out in (32)):

  38) A B G

  In this linear string, A immediately precedes B, because A precedes B and there is nothing in between them.

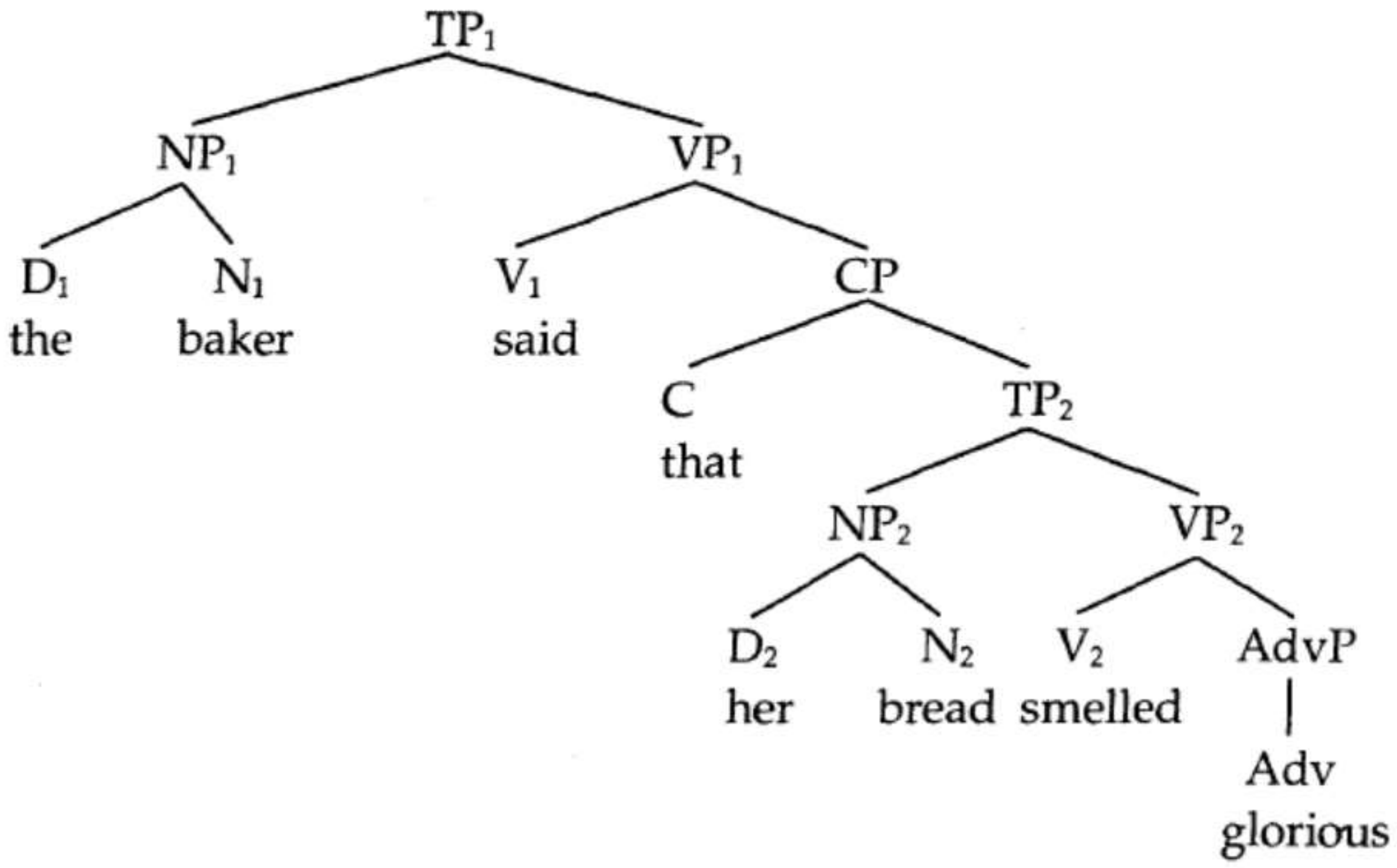
- Contrast this with (39):

  39) A G B

  In this string, A does *not* immediately precede B. It does precede B, but G intervenes between them, so the relation is not immediate.

Precedence exercise:

Q. Study the tree carefully and answer the following questions.



Q. What node precedes what? List them all.

1) List all the nodes that precedes each of the following items (For some elements there may be nothing that precedes them)

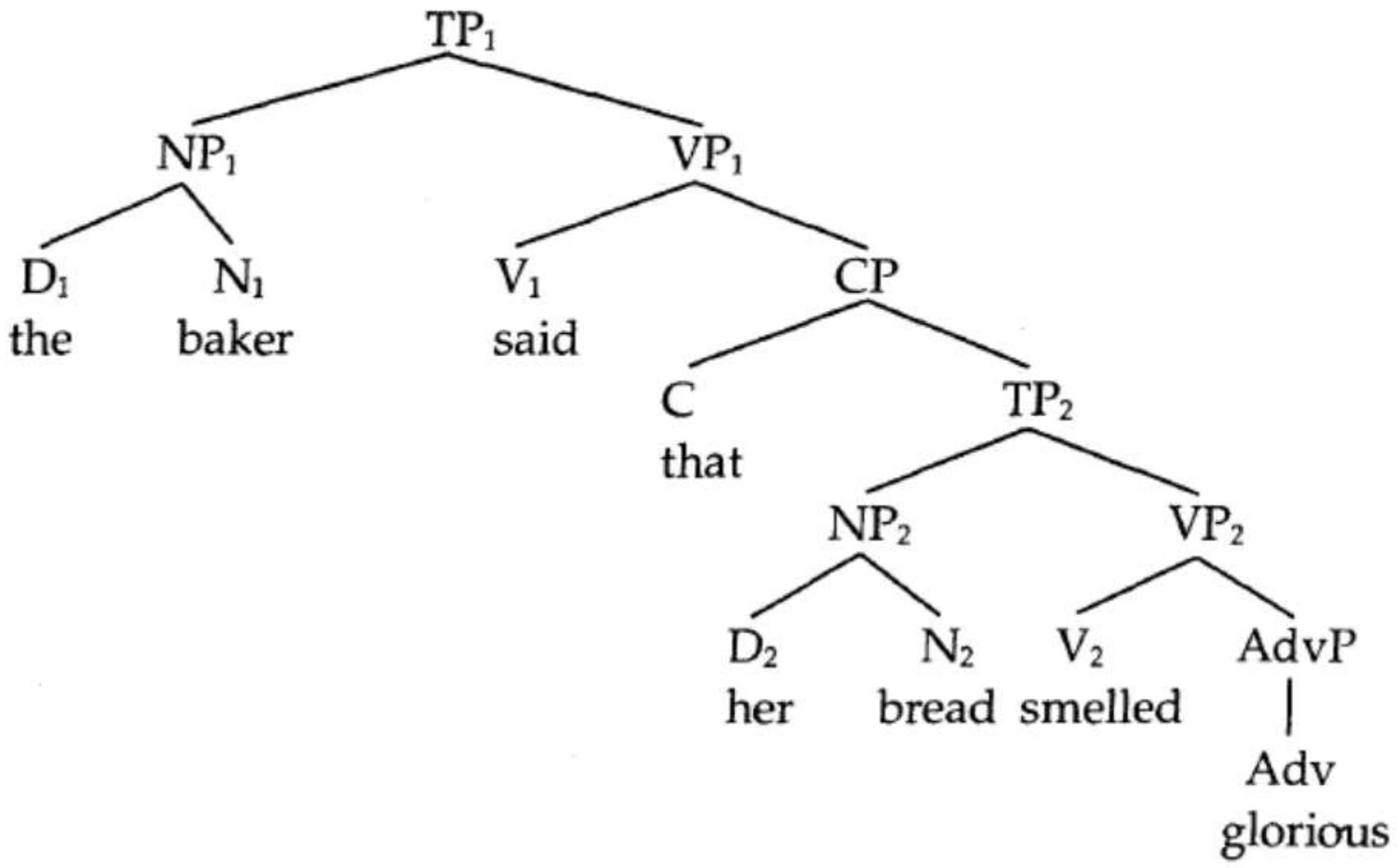
a) D_1 *the* b) D_2 *her* c) N_1 *baker* d) N_2 *bread* e) V_1 *said* f) V_2 *smelled* g) Adv *glorious*

h) C *that* i) TP_1 (if there are any) j) TP_2 k) NP_1 l) NP_2 m) VP_1

n) VP_2 o) CP p) AdvP

Immediate Precedence exercise:

Q. Study the tree carefully and answer the following questions.



1) List all the nodes that immediately precedes each of the following items (For some elements there may be nothing that immediately precedes them)

a) D_1 *the* b) D_2 *her* c) N_1 *baker* d) N_2 *bread* e) V_1 *said* f) V_2 *smelled* g) Adv *glorious*

h) C *that* i) TP_1 (if there are any) j) TP_2 k) NP_1 l) NP_2 m) VP_1

n) VP_2 o) CP p) AdvP

4. C-Command

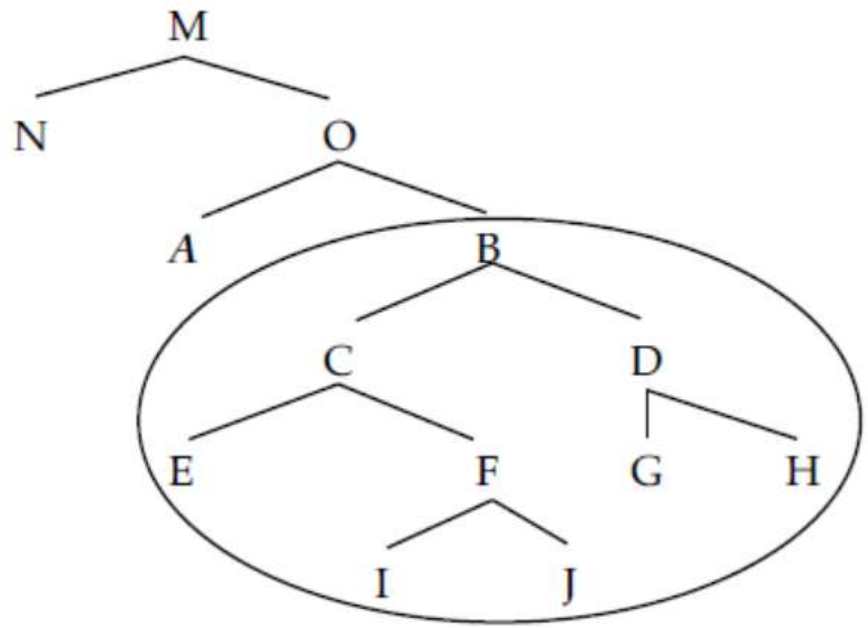
- Perhaps the most important of the structural relations is the one we call ***c-command***.
- Although c-command takes a little getting used to, it is actually the most useful of all the relations. In the next chapter, we'll look at the phenomenon of ***binding***, which makes explicit reference to the c-command relation.
- C-command is defined intuitively in (40) and more formally in (41):

40) *C-command (informal):* A node c-commands its sister$ and all the daughters (and granddaughters and great-granddaughters, etc.) of its sisters.

41) *C-command (formal):* Node A c-commands node B if every node dominating A also dominates B, and neither A nor B dominates the other, and A ≠ B.

- Look at the tree in (42). The node A c-commands all the nodes in the circle. It doesn't c-command any others:

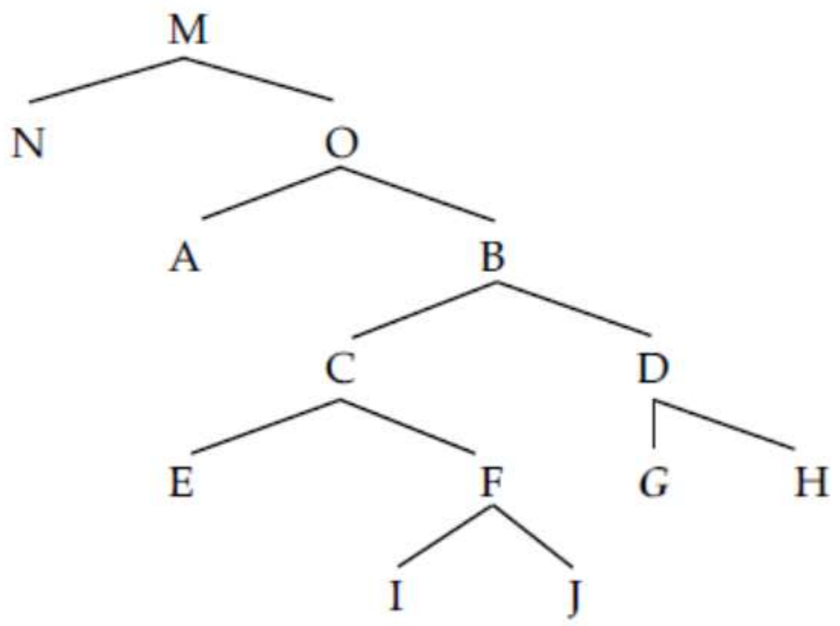
42)



- That is, A c-commands its sister (B) and all the nodes dominated by its sister (C, D, E, F, G, H, I, J).

- Consider now the same tree without the circle, and look at the nodes c-commanded by G:

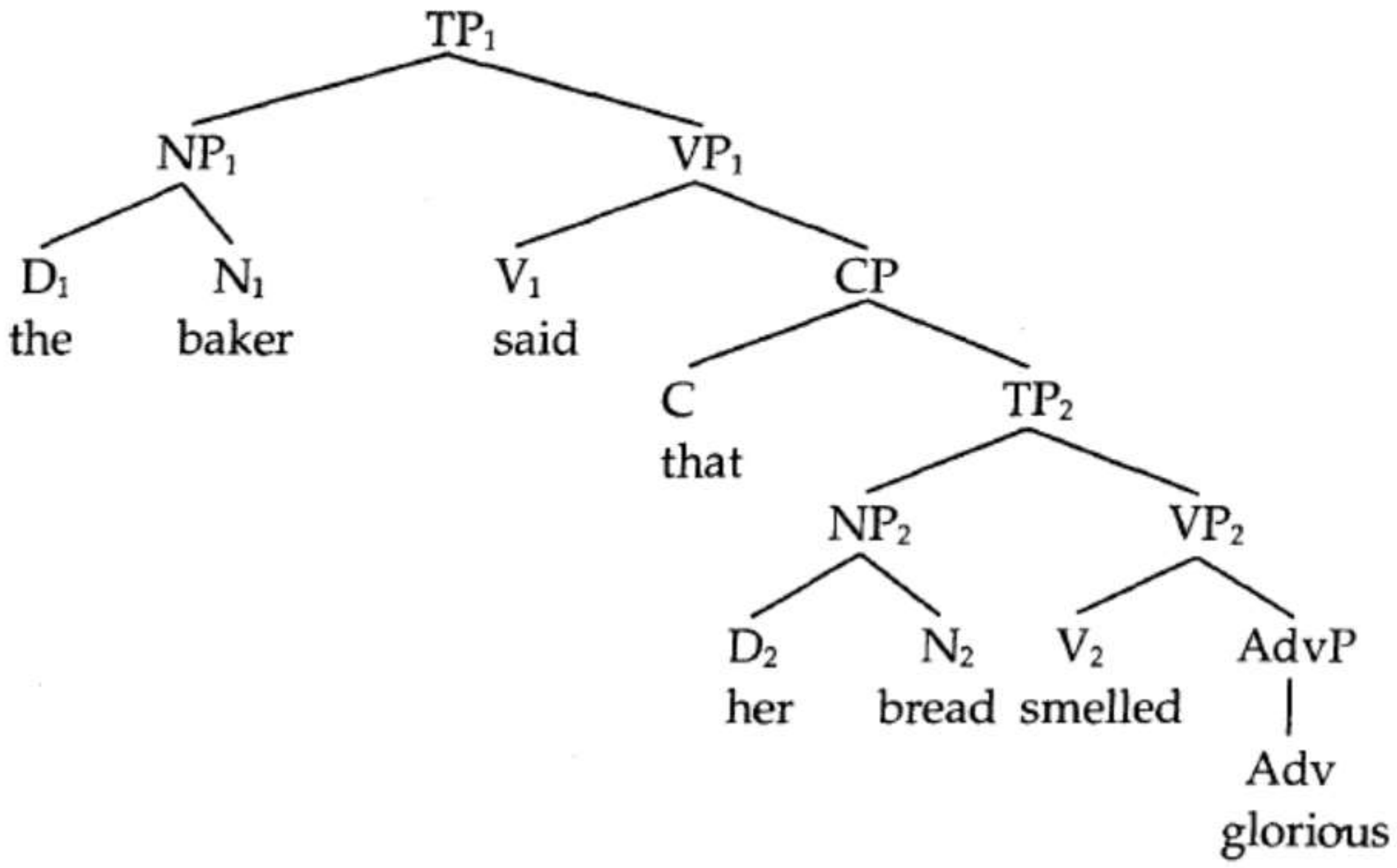
43)



- G c-commands *only* H (its sister).
- Notice that it does not c-command C, E, F, I, or J.
- C-command is a relation that holds among sisters and among aunts and their nieces and the descendants of their nieces.
- It *never* holds between cousins or between a mother and daughter.

C-command exercise:

Q. Study the tree carefully and answer the following questions.



1) List all the nodes that the following nodes c-command:

(For some nodes may not c-command any nodes)

a) D_1 *the* b) D_2 *her* c) N_1 *baker* d) N_2 *bread* e) V_1 *said* f) V_2 *smelled* g) Adv *glorious*

h) C *that* i) TP_1 (if there are any) j) TP_2 k) NP_1 l) NP_2 m) VP_1

n) VP_2 o) CP p) AdvP

2) What nodes c-command TP_2?

3) What nodes c-command NP_1?

4) What nodes c-command C?

- There are various kinds of c-command.
- The **first** of these is when two nodes c-command one another. This is called ***symmetric c-command*** and is defined in (44):

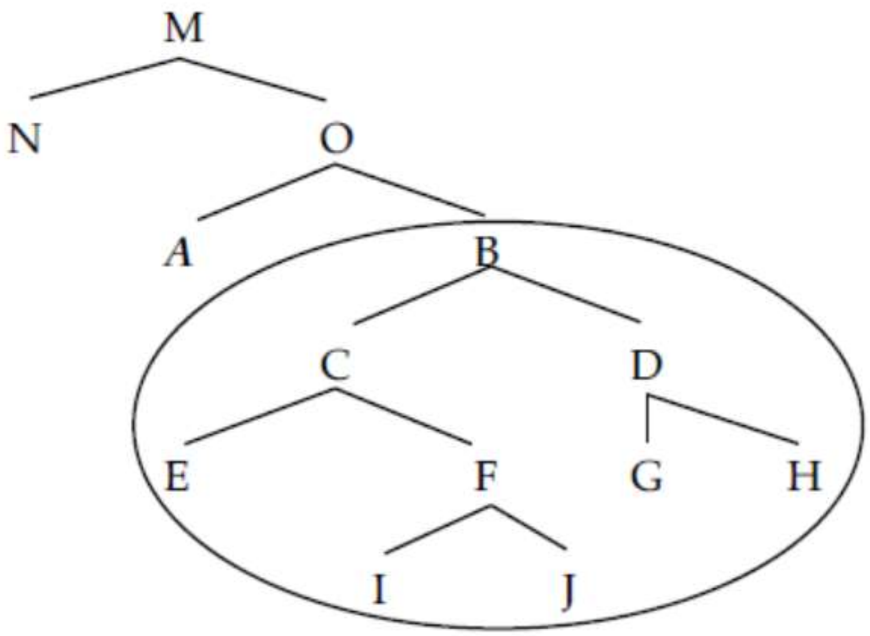
44) ***Symmetric c-command****:* A symmetrically c-commands B if A c-commands B *and* B c-commands A.

- This relation holds only between sisters.

- The **other** kind of c-command is the kind that holds where an aunt c-commands her nieces and the descendants of her nieces. This is called (unsurprisingly) ***asymmetric c-command****:*

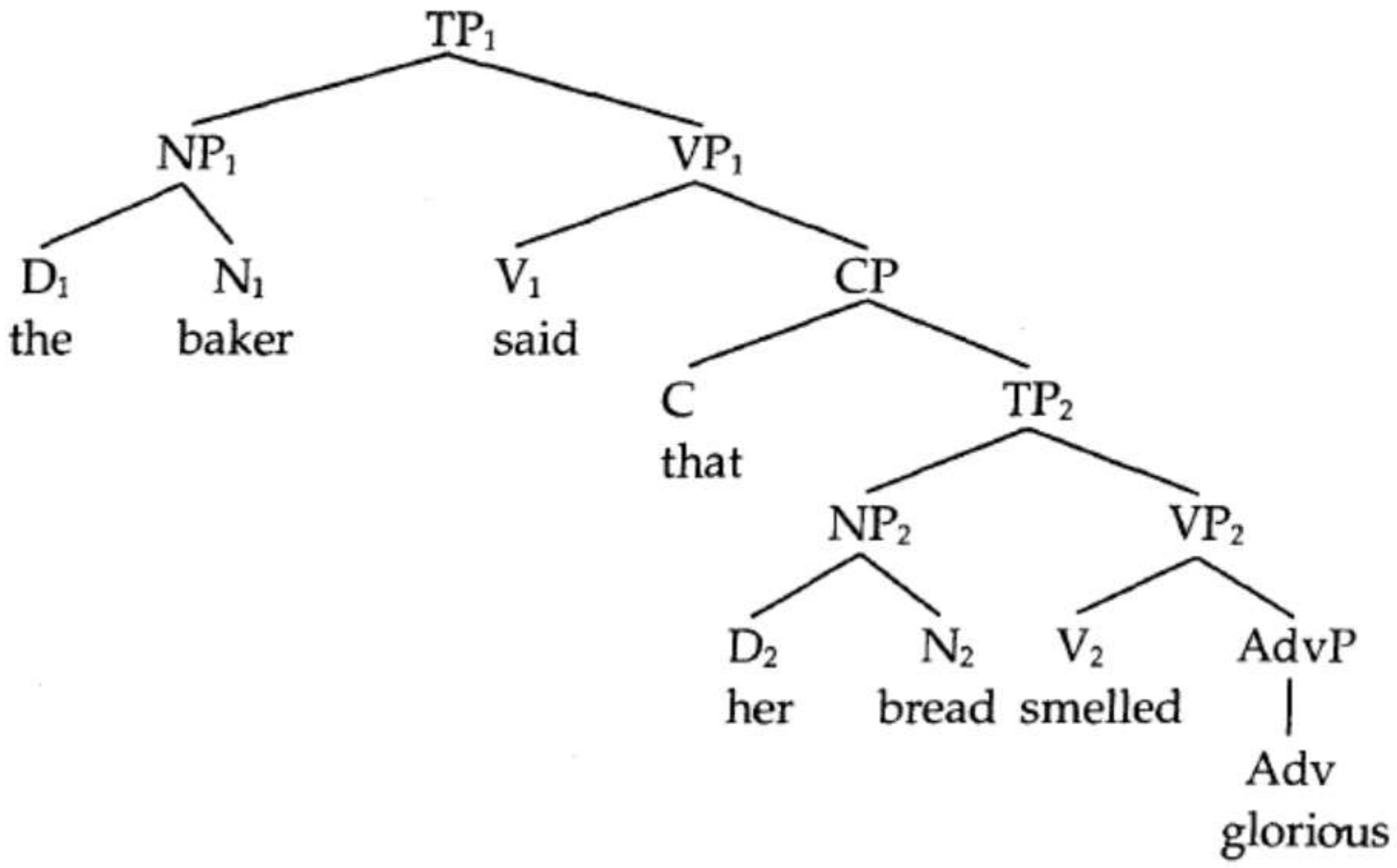
45) ***Asymmetric c-command:*** A asymmetrically c-commands B if A c-commands B but B does *not* c-command A.

- Consider again the tree in (42 *but repeated below for convenience*); N and O symmetrically c-command each other (as do all other pairs of sisters).
- However, N asymmetrically c-commands A, B, C, D, E, F, G, H, I, and J, since none of these c-command N.



Symmetric and Asymmetric C-command exercise:

Q. Study the tree carefully and answer the following questions.



1) List all the nodes that the following nodes symmetrically c-command (if there are any):

a) D_1 *the* b) D_2 *her* c) N_1 *baker* d) N_2 *bread* e) V_1 *said* f) V_2 *smelled* g) Adv *glorious*

h) C *that* i) TP_1 j) TP_2 k) NP_1 l) NP_2 m) VP_1

n) VP_2 o) CP p) AdvP

2) List all the nodes that the following nodes asymmetrically c-command (if there are any):

a) D_1 *the* b) D_2 *her* c) N_1 *baker* d) N_2 *bread* e) V_1 *said* f) V_2 *smelled* g) Adv *glorious*

h) C *that* i) TP_1 j) TP_2 k) NP_1 l) NP_2 m) VP_1

n) VP_2 o) CP p) AdvP

3) What nodes asymmetrically c-command V_2?

4) What nodes symmetrically c-command NP_1?

5) What nodes asymmetrically c-command C?

6) What nodes symmetrically c-command C?

- Just as we had local (immediate) versions of **dominance** and **precedence**, there is a local version of c-command. This is typically called ***government*** (rather than immediate c-command).

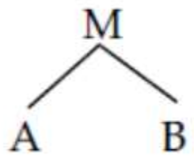
- So for **dominance**, we said, some node A immediately dominates B provided there is no intermediate node G that A dominates and that dominates B.
- In (46a) there is no node between A and B, so A immediately dominates B. In (46b), by contrast, G is in between them, so A does not immediately dominate B.

46)

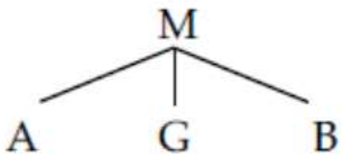
- The same idea played a role in **precedence**.
- In (47a), A immediately precedes B because there is nothing between them; in (47b) A precedes B, but it doesn't immediately precede B, because G intervenes.

47)

a)



b)

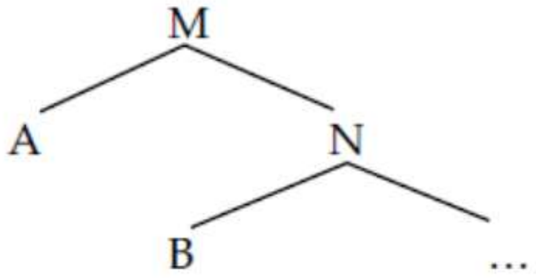


- Government is similarly defined:

  48) ***Government*** *(first version):* Node A governs node B if A c-commands B, and there is no node G, such that G is c-commanded by A and G asymmetrically c-commands B.

  To see this at work, look at the tree in (49):

  49)

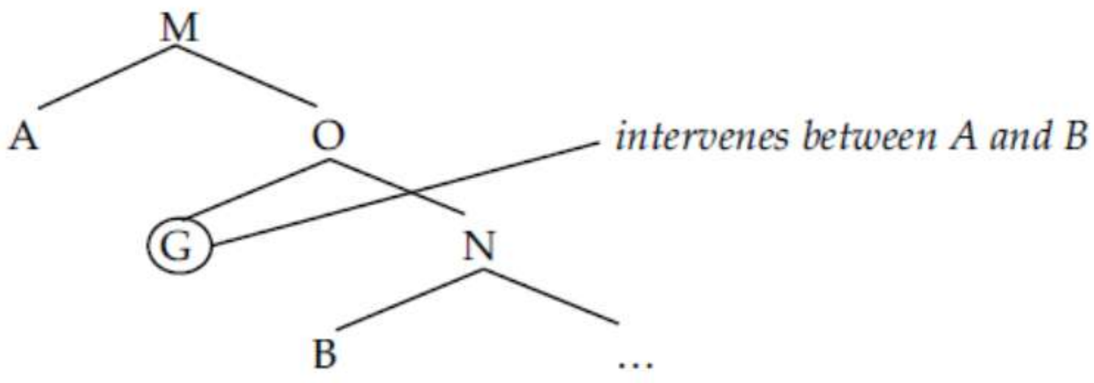


- In this tree, A governs B.
- It c-commands B, and there is no node that c-commands B that A also c-commands.

(You should note that A also governs N under this definition, A c-commands N, and there is no node that N c-commands that also c-commands A.
The reverse is also true: N governs A because the relationship between A and N is symmetric c-command.
B does not govern A, because B does not c-command A.)

Contrast this with the tree in (50):

50)

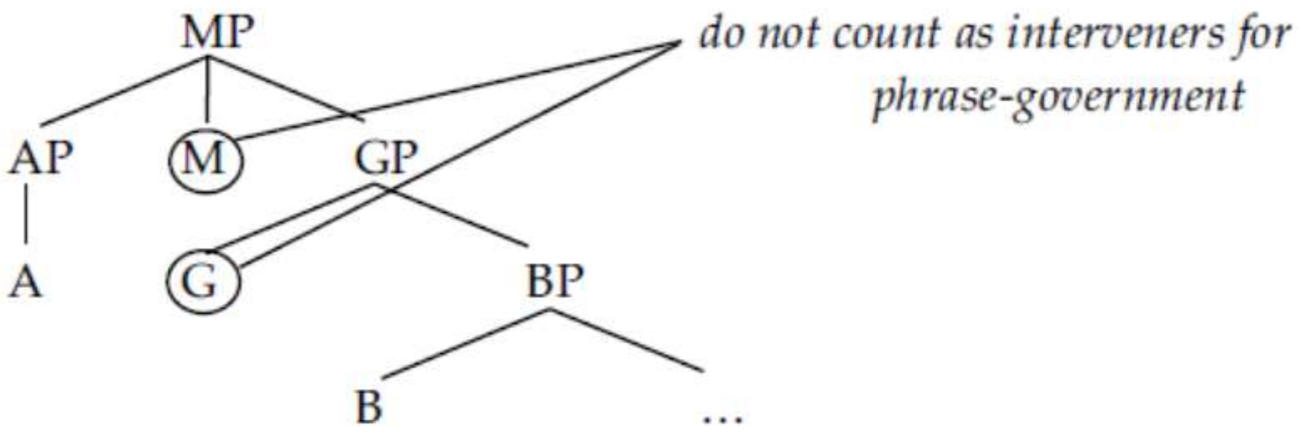


Here A does not govern B, because the node G intervenes (or more precisely, A c-commands G and G c-commands B, thus violating the definition).

- Government is often "relativized" to the particular kind of element that's doing the government. For example, if the governor (the element doing the governing) is a phrase (an NP, a VP, etc.), then what count as interveners are only other phrases, not heads like N, V, etc.

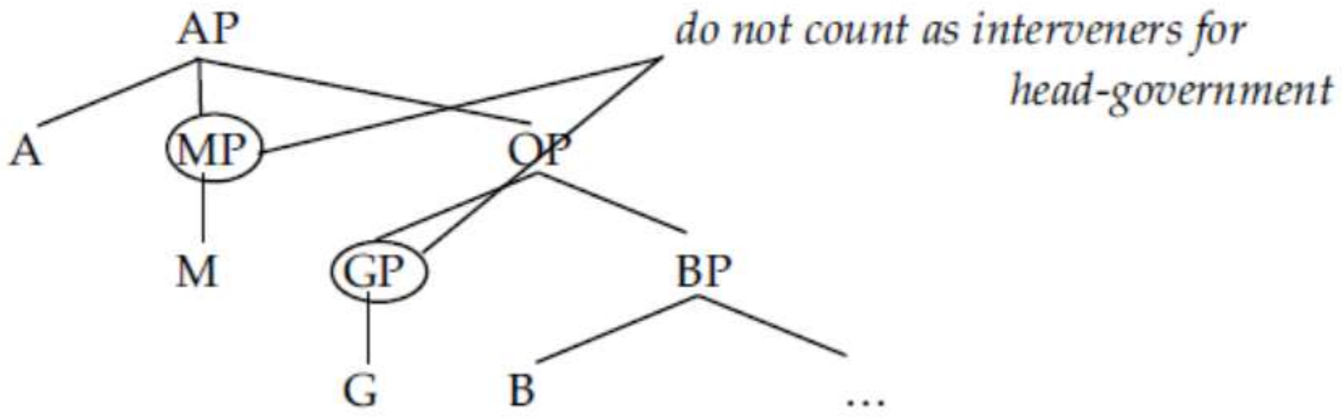
- In (51) the AP *phrase-governs* B. G and M don't count as interveners, even though they both are c-commanded by AP and they both c-command B.
- This is because they are not phrases – they are heads. GP and BP don't count as interveners either, because they don't command B; they dominate it.

51)



- Similarly, if the governor is a head *(head-government)*, then phrasal interveners don't count:

52)



In (52), MP and GP do not count as interveners for A head-governing B because they are phrases.
M and G don't count because they don't c-command B. With this in mind, we can revise the definition:

53) ***Government***

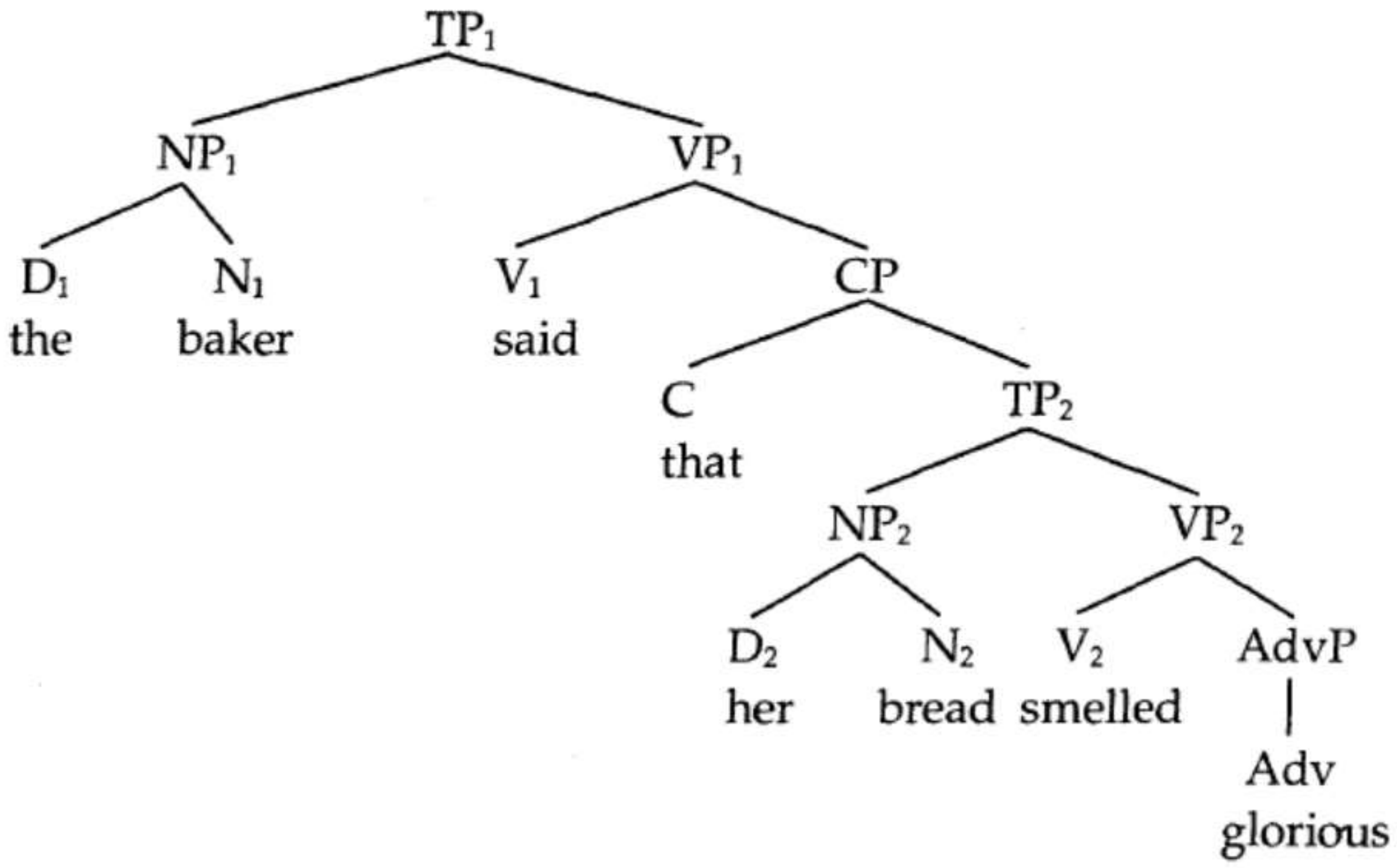
Node A governs node B if A c-commands B and there is no node G such that G is c-commanded by A and G asymmetrically c-commands B.

a) ***Phrase-government:*** If A is a phrase, then G must also be a phrase.
b) ***Head-government.*** If A is a head (word), G must also be a head.

- In recent years, government has to a greater or lesser degree fallen out of fashion.
- Instead local relations previously linked to government are often determined by what is called the specifier-head relation. However, it is important to know what government is, because many influential papers in syntax refer to this relation.

Government exercise:

Q. Study the tree carefully and answer the following questions.



1) Does NP_1 govern VP_2? Why or why not?
2) Does NP_1 govern C *that?* Why or why not?
3) What nodes does N_1 govern?
4) Does V_1 head-govern V_2? Why or why not?
5) Does NP_1 phrase-govern AdvP? Why or why not?
6) Does VP_2 phrase-govern N_2? Why or why not?

Q. Draw a single tree with the following properties:

a) R is the root node
b) B is a terminal node and precedes all other terminal nodes
c) C dominates B
d) C sister-precedes D
e) {F, G, H} are exhaustively dominated by D
f) F asymmetrically c-commands G and H
g) E is immediately dominated by D
h) F precedes E
i) G sister precedes H

5. GRAMMATICAL RELATIONS

- In addition to the structural relations that hold among items in a tree, there are some traditional grammatical terms that can be defined structurally. These are useful terms, and we will frequently make reference to them. We call these *grammatical relations*.
- Technically speaking, grammatical relations are not structural relations.
- Some theories of grammar (for example, Lexical-Functional Grammar and Relational Grammar) posit primitive grammatical relations (meaning they are not structurally defined).
- In the approach we are developing here, however, grammatical relations are defined structurally; that is, they are defined in terms of the tree.

- In English the subject is always the NP or CP that appears before the verb or auxiliary:

  54) a) *The puppy* licked the kitten's face.
  b) *It* is raining.
  c) *Gabby* feels fine.
  d) *The kitten* was licked.
  e) *That Bill's breath smells of onions* bothers Erin.

- Notice that the definition of subject is not a semantic one.
- It is not necessarily the doer of the action.
- In (54c) for example, Fred is not deliberately feeling fine.
- In sentence (54d), the kitten is the one being licked, not the licker.
- Different semantic types of noun phrases appear to be allowed to function as the subject.

- There is a straightforward structural definition of the *subject:*

  55) *Subject (preliminary):* NP or CP daughter of TP

- Next, we have the *direct object* of the verb and the *object of a preposition*. Examples of these are seen in (56) and (57), respectively:

  56) *Direct object*

  a) Susan kissed *the clown's nose*.

  b) Cedric danced *a jolly jig*.

  c) Dale said *that the lawn was overgrown*.

  57) *Object of a preposition*

  a) Gilgamesh cut the steak with *a knife*.

  b) We drove all the way to *Buenos Aires*.

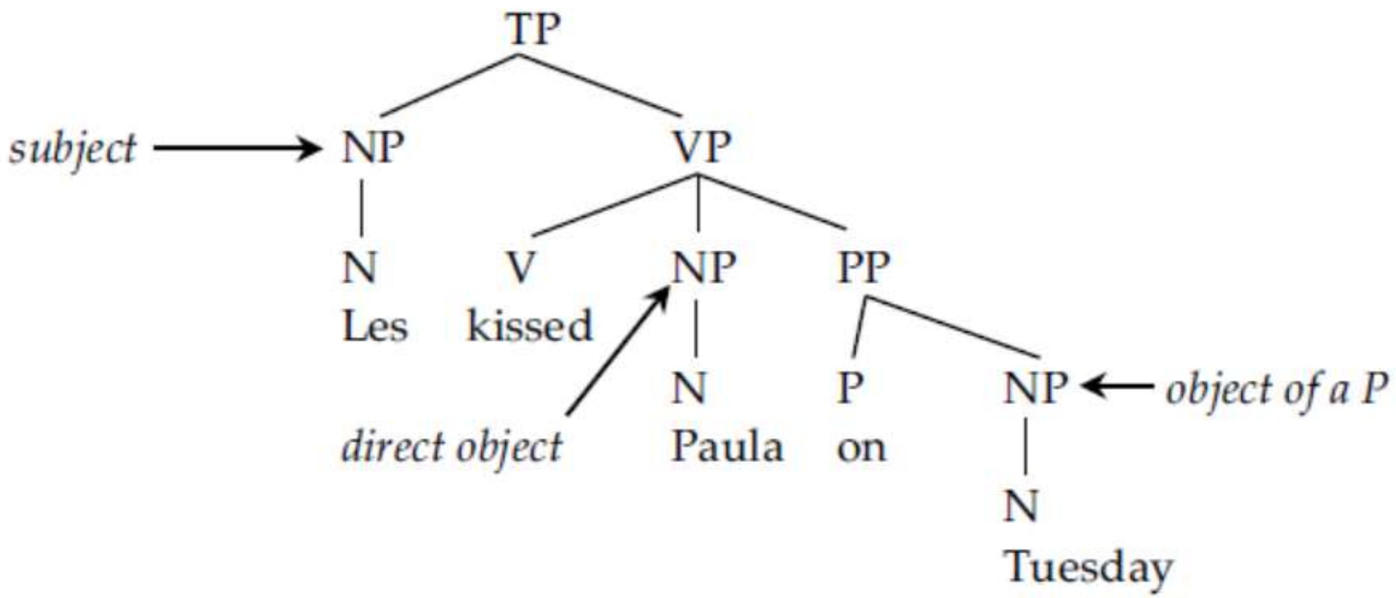
Preliminary definitions of these are given in (58) and (59); again, we will have reason to revise these later.

58) *(Direct) object (preliminary):* NP or CP daughter of a VP

59) *Object of preposition:* NP daughter of PP

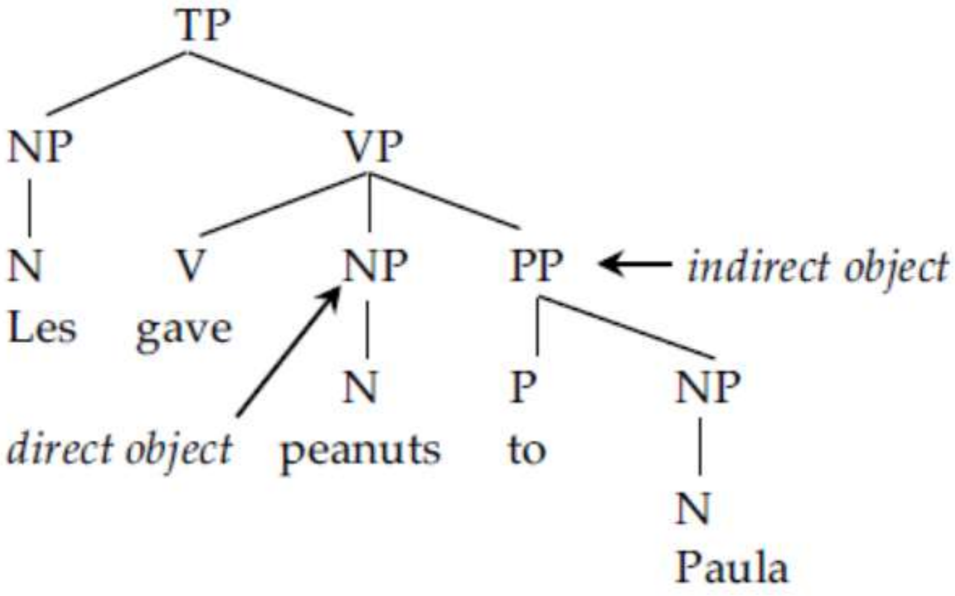
- To see these definitions at work, consider the following tree.
- The NP *Paula* is the daughter of TP and is thus the subject.
- The NP *Les* is a daughter of the VP headed by the transitive verb *kissed*, so *Les* is the direct object.
- *Tuesday* is the NP daughter of a PP, thus the object of a preposition.

60)



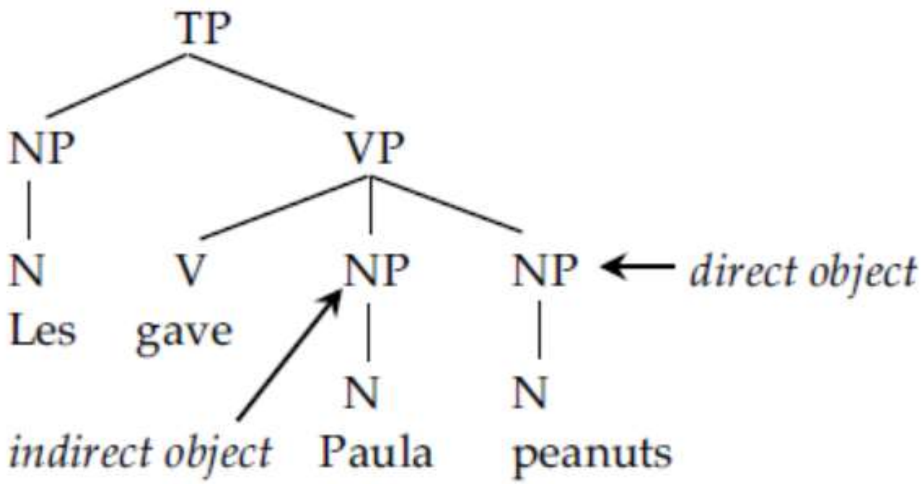
- In addition to direct objects, when you have a ditransitive verb like *give* or *put* you also have an indirect object. The *indirect object* in English shows up in two places. It can be the PP that follows the direct object:

61)



- It can also be the *first* NP after the verb when the verb takes two NPs:

62)



Notice that the direct object is the second of the two NPs, roughly the reverse of the tree in (61).

- This means complicating our definitions somewhat:

63) *Direct Object (second pass):*

a) The NP or CP daughter of VP ($V[_{NP\_\_NP}]$, $v[_{NP\_\_CP}]$ and $v[_{NP\_\_NP\ PP]}$).

b) The NP or CP daughter of VP that is preceded by an NP daughter of VP ($V_{[NP\ \_\_\ NP\ \{NP/CP\}]}$).

64) *Indirect Object (preliminary):*

a) The PP daughter of VP immediately preceded by an NP daughter of VP. ($V_{[NP\_\_\ NP\ PP]}$)

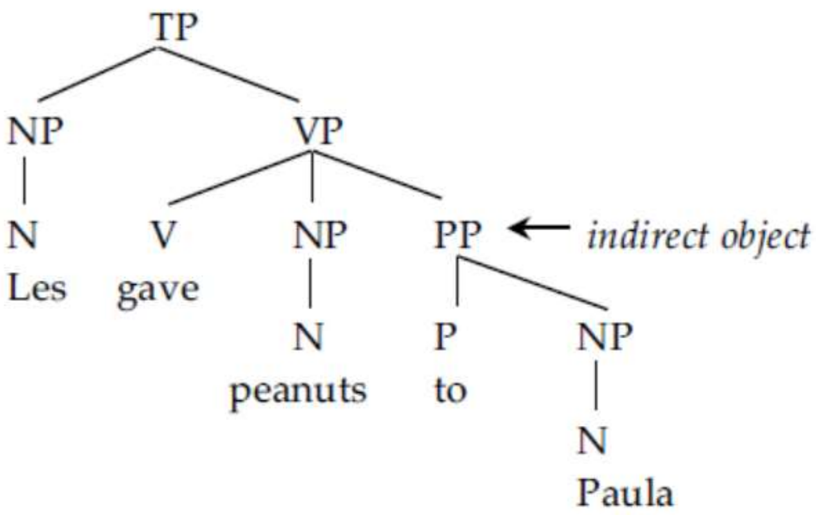
b) The NP daughter of VP immediately preceded by V (i.e., the first NP daughter of VP). ($V_{[NP\ \_\_NP\ \{NP/CP\}]}$)

- In addition to subjects, objects, and indirect objects, you may also occasionally see reference made to *obliques*.
- In English, obliques are almost always marked with a preposition.
- The PPs in the following sentence are obliques:

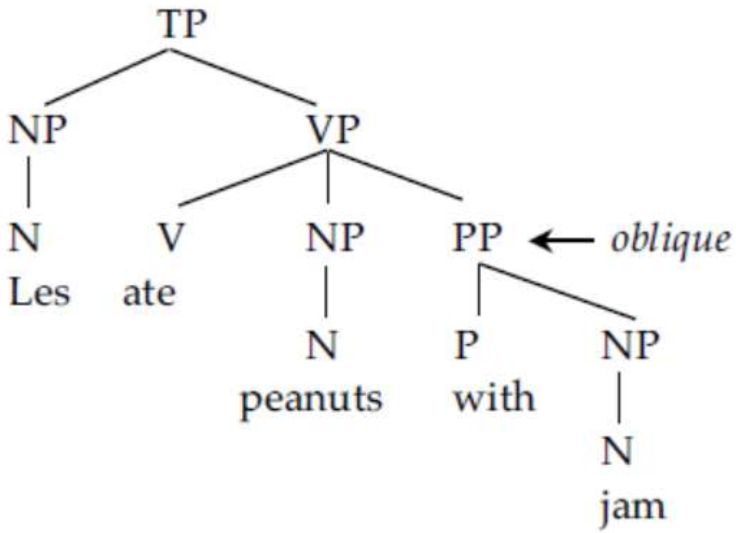
  65) Alvina tagged Moises [$_{PP}$ *with a regulation baseball*] [$_{pp}$ *on Tuesday*].

- Notice that obliques can structurally show up in the same position as indirect objects (compare (66a) to (66b)).
- The difference between the two is in whether the PP is part of the argument structure of the verb or not.
- If the verb is of type $V_{[NP\_NP\ PP]}$ like *give*, then the PP is an indirect object, but if the verb is of type $V_{[NP\_NP]}$ (where the PP isn't specified by the feature), like *eat*, then the PP is an oblique.

66a)



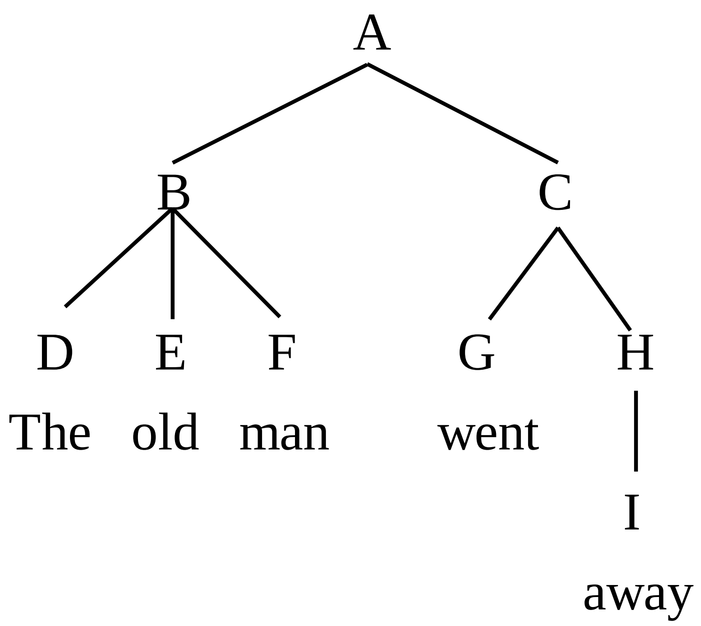
66b)

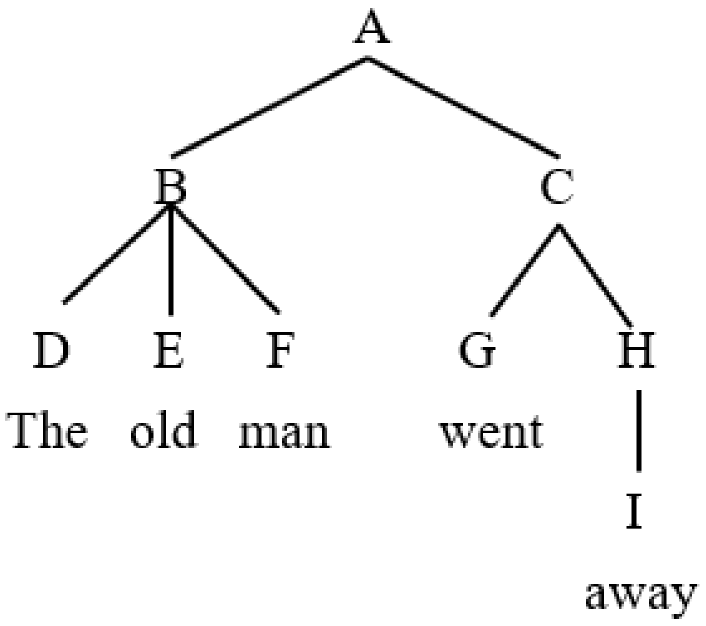


Reference

Carnie, A. (2021). *Syntax: A Generative Introduction* (Fourth Edition). Wiley Blackwell.

here

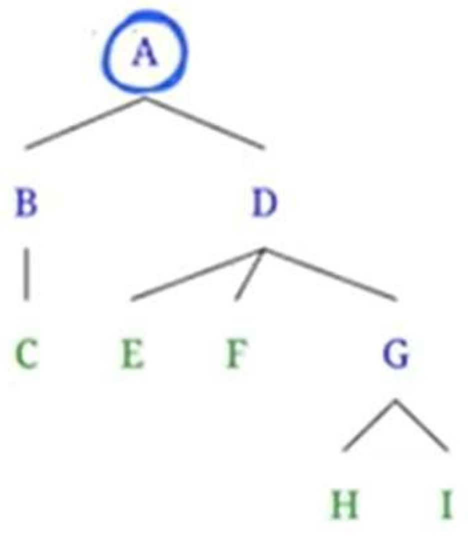

A
B
C
D
E
F
G
H
The
old
man
went
I
away


A
B
C
D
E
F
G
H
The
old
man
went
I
away

Q.

Dominance is related to the "height" of nodes and the relationships with other nodes.

- **Dominance**:
  X dominates Y iff X is higher than Y in the tree.
- **Immediate Dominance**:
  X immediately dominates Y iff X is the mother of Y.
- **C-Command**:
  X c-commands Y iff X is a sister to Y or X is a sister to a node dominating Y.



Q.

Q.

Q.