chapter 4

Structural Relations

0. Introduction

In chapter 3, 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. Take a careful look at any tree in the last chapter and you'll notice that it is a collection of labels and lines; within this collection of labels there is an organization. In particular, various parts of the tree are organized hierarchically with respect to one another. A collection of lines and labels with an internal organization like syntactic trees is a geometric object. It isn't a geometric object like a circle or a square, but nonetheless it has bits that are spatially organized with respect to one another. If syntactic trees are geometric objects, they can be studied and described mathematically – the focus of this chapter. This chapter differs from all the others in this book. You won't see many sentences or phrases here, and there is very little data. This chapter is about the purely formal properties of trees. But don't think you can skip it. The terminology we develop here is a fundamental part of syntactic theory and will play an important role in subsequent chapters.

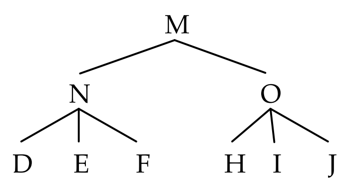
Why Study the Geometry of Trees?

It is worth considering whether it is necessary to concern ourselves with the mathematics of tree diagrams. There are actually two very good reasons why we should do this. 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 chapter we were only able to give a vague definition of the term *constituent*. In this chapter, we'll be able to give a precise description. Second, it turns out that there are many syntactic phenomena that make explicit reference to the geometry of trees. One of the most obvious of these refers to anaphors. Anaphors can only appear in certain positions in the geometry of the tree. The distribution of anaphors and other types of nouns is the focus of the next chapter.

1. THE PARTS OF A TREE

Let's start with a very abstract tree drawing:

1)



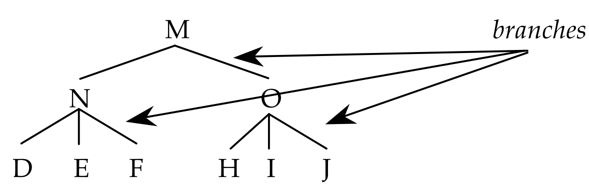
This tree would be generated by the rules in (2):

2) M → N O
N → D E F
O → H I J

You can check this by applying each of the rules to the tree in (1). I'm using an abstract tree here because I don't want the content of each of the nodes to interfere with the underlying abstract mathematics. (But if you find this confusing, you can substitute TP for M, NP for N, VP for O, etc., and you'll see that this is just a normal tree.) 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 with two or more branches below it is said to be ***branching***; a node that has a single branch below it is said to be ***non-branching***.

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 chapter, 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. (The term root is a little confusing, but try turning the trees upside down and you'll see that they actually do look like a tree (or a bush at least). In most trees we looked at in the last chapter, the root node is 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***.

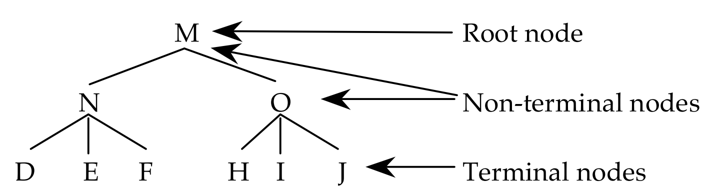
8) *Terminal node (preliminary):* Any node with no branch underneath it.

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 chapter, 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.

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***.

2. DOMINATION

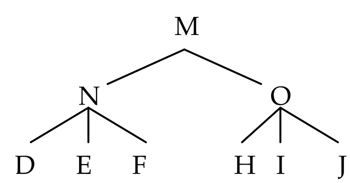
2.1 Domination

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*[1]: 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, J. O does not dominate F, as you can see by virtue of the fact that there is no branch connecting them.

12)



[1] The definition given here is actually for proper domination (an irreflexive relation). Simple domination is usually reflexive (nodes dominate themselves). For the most part linguists are interested in proper domination rather than simple domination, and they use the term "domination" to mean "proper domination" as we do here. Domination is sometimes also called ***dominance***.

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]) contains 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.

You now have enough information to try General Problem Sets 1 & 2

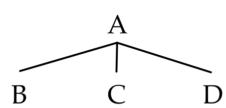
2.2 Exhaustive Domination

In the last chapter, 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***:

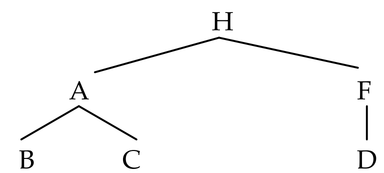
14) *Exhaustive domination*: Node A exhaustively dominates a *set* of terminal nodes {B, C, ..., D}, provided it dominates all the members of the set (so that there is no member of the set that is not dominated by A) *and* there is no terminal node G dominated by A that is not a member of the set.

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

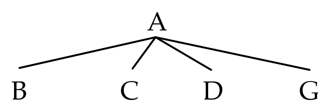
15)



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 (15). 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 (15) 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 immediately dominated by A. As such the set {B, C, D} is *not* exhaustively dominated by A. The reverse situation is seen in (17):

17) 

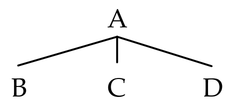
While it is the case that in (17), B, C, and D are all immediately dominated by A, there is also the node G, which is not a member of the set {B, C, D}, so the set {B, C, D} is not exhaustively dominated by A (although the set {B, C, D, G} is). On a more intuitive level, exhaustive domination holds between 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 (16), that set does not form a constituent, nor does it form a constituent in (17) (although the set is part of a larger constituent in that tree). In (17), there is no sense in which B, C, 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) 

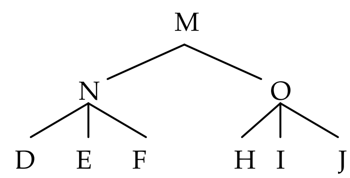
Here we have the constituent A, which exhaustively dominates the set {B, C, D}. Each member of this set is called a "constituent of A." So B is a constituent of A. "Constituent of" boils down to domination. A dominates B therefore B is a constituent of A:

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

You now have enough information to try General Problem Set 3

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 domination.*** 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 domination. 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 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. (A node that is a mother.)

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

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.

You now have enough information to try General Problem Set 4

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 $\rightarrow$ 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***.[2] 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 → 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:

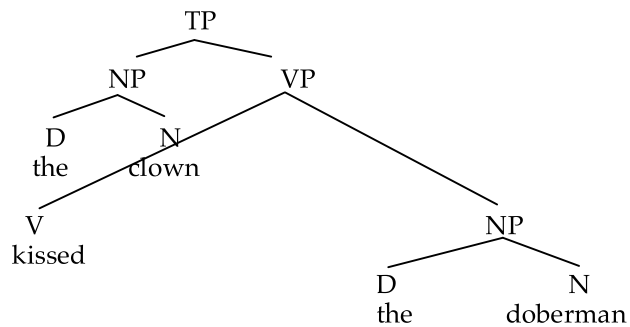
32) *Precedence*: Node A precedes node B if and only if neither A dominates B nor B dominates A *and* A or some node dominating A sister-precedes B or some node dominating B.

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, domination 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 domination. 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 seems like an overly complex way to say "to the left," but there is a good reason we phrase it this. This has to do with the fact that the terminals of a tree don't float out in space, 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:

[2] Thanks to Dave Medieros for helpful discussion of these notions.

33)



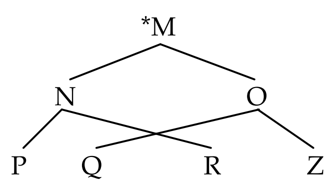
In this sloppily drawn tree, the verb *kissed* actually appears to the *left* of the noun *clown*. However, we wouldn't want to say that *kissed* precedes *clown*; this is clearly wrong. The sentence is said "The clown kissed the doberman," where *kissed* follows *clown*. We guarantee this ordering by 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 below 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.

You now have enough information to try General Problem Set 5 and Challenge Problem Set 1

Just as in the domination relation, where there is the special local definition called "immediate domination," 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.

You now have enough information to try General Problem Set 6

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):

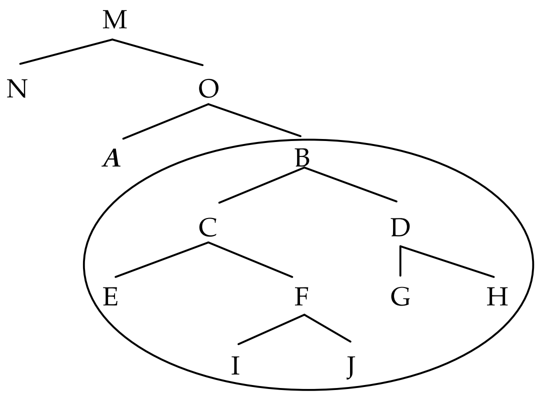
40) *C-command (informal):* A node c-commands its sisters and all the daughters (and granddaughters and great-granddaughters, etc.) of its sisters.

41) *C-command (formal):* Node A c-commands node B if every[3] node dominating A also dominates B, and neither A nor B dominate the other.

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

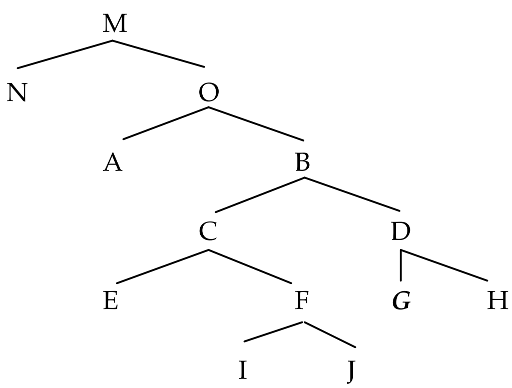
[3] The usual requirement on c-command is that every ***branching*** node dominating A also dominate B. This additional branching requirement isn't necessary given the irreflexive definition of domination (i.e. proper domination) that we've given above. However, students may run into the branching definition in other works if other definitions of domination are used.

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 between sisters and aunts and nieces. It *never* holds between cousins or between a mother and daughter.

You now have enough information to try General Problem Set 7 and Challenge Problem Set 2

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 between an aunt and 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); 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.

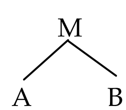
You now have enough information to try General Problem Set 8

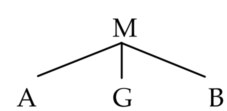
Just as we had local (immediate) versions of domination and precedence, there is a local version of c-command. This is typically called ***government***[4] (rather than immediate c-command). There are a number of different definitions for government. If you look back at our definitions for immediate precedence and immediate domination, you'll see that in both cases the locality (i.e., the closeness) of the relationship was defined by making reference to a potential intervening category. So for domination, some node A immediately dominates B another 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) a)


b)


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)


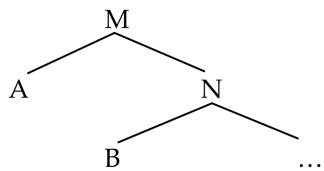
Government is similarly defined:

---

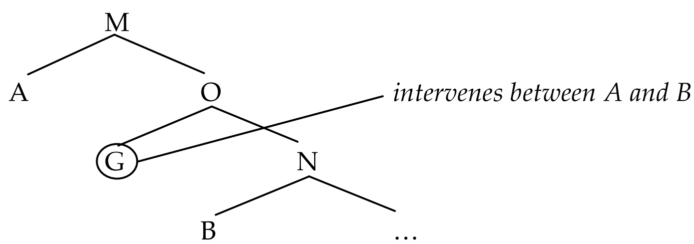
[4] Technically speaking, government isn't just immediate c-command, it also expresses a licensing relationship (that is it has or had the special status of a constraint on the grammar). In this book, this licensing function isn't going to be used, so we're going to concentrate on the structural relationship part of the definition only.

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 c-commands B.

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

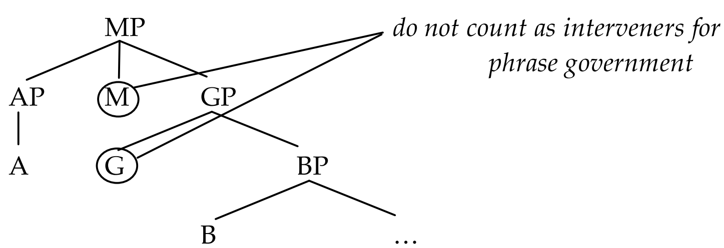
49)


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-command B, thus violating the definition).

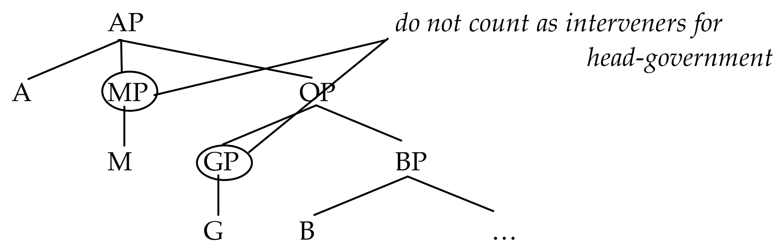
Government is often "relativized" to the particular kind of element that's doing the government. For example, if the governor (the element doing the governing is a phrase (an NP, a VP, etc.), then what counts as an intervener are only other phrases, not heads like N, V, etc.

51)


In (51) the AP ***phrase-governs***[5] 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.

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.[6] 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 c-commands B.

- ***Phrase-government***: If A is a phrase, then the categories that count for G in the above definition must also be phrases
- ***Head-government***: If A is a head (word), then the categories that count for G in the above definition must also be heads.

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 if you read many influential papers in syntax they will refer to this relation.

You now have enough information to try General Problem Set 9

---

[5] In the syntactic literature, this is more usually called ***antecedent government*** (which has an additional constraint called coindexing on it and is defined over particular categories – so NP antecedent governs another coindexed NP, provided there is no intervening c-commanding NP that also is c-commanded by the first). This is a refinement that we won't pursue here because it is rarely used anymore.

[6] These don't c-command B only if the branching requirement on c-command does not hold.

5. GRAMMATICAL RELATIONS

In addition to the structural relations that hold between items in a tree, there are also 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) *Fred* 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[7] 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

In later chapters, we will have cause to refine this definition somewhat, but for now, this will do.

Next we have the ***direct object*** of the verb and the ***object of a preposition***. Examples of these are seen in (53) and (54) 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*.

[7] In chapter 8, we will look at different semantic types of noun phrases. These types are called ***thematic relations***.

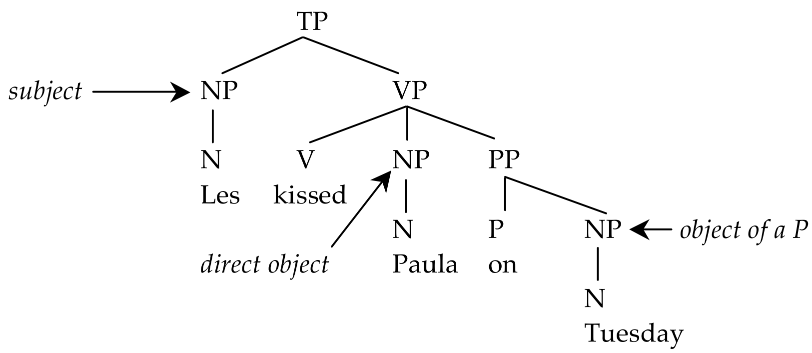
Preliminary definitions of these are given in (58) and (59), again we will have reason to revise these in later chapters.

58) *(Direct) object (preliminary)*: NP or CP daughter of a VP headed by a transitive verb

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

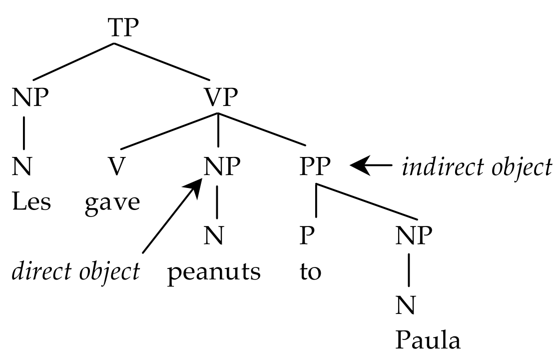
To see these definitions at work consider the following tree. The NP *Les* is the daughter of TP, and is thus the subject. The NP *Paula* is a daughter of the VP headed by the transitive verb *kissed*, so *Paula* is the direct object. *Tuesday* is the NP daughter of a PP, thus the object of a preposition.

60)



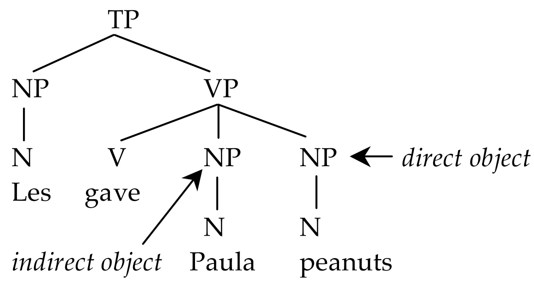
In addition to direct objects, when you have a ditransitive verb like give or put, you also have an indirect object. Indirect objects in English come of several types in terms of the types of arguments they take. The two most common types are the direct object which we preliminarily defined above and 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) With verbs of type $V_{[NP\_\_NP]}$, $V_{[NP\_\_CP]}$ and $V_{[NP\_\_\ NP\ PP]}$, the NP or CP daughter of VP
  b) With verbs of type $V_{[NP\ \_\_\ NP\ \{NP/CP\}]}$, An NP or CP daughter of VP that is preceded by another NP daughter of VP. (i.e., the second NP daughter of VP)

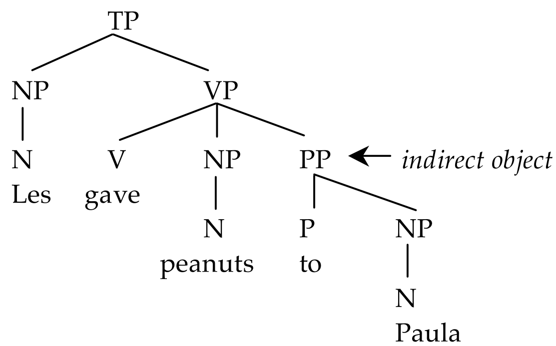
64) *Indirect Object (preliminary):*
  a) With verbs of type $V_{[NP\_\_\ NP\ PP]}$, the PP daughter of VP immediately preceded by an NP daughter of VP.
  b) With verbs of type $V_{[NP\ \_\_\ NP\ \{NP/CP\}]}$, the NP daughter of VP immediately preceded by V (i.e., the first NP daughter of VP).

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:

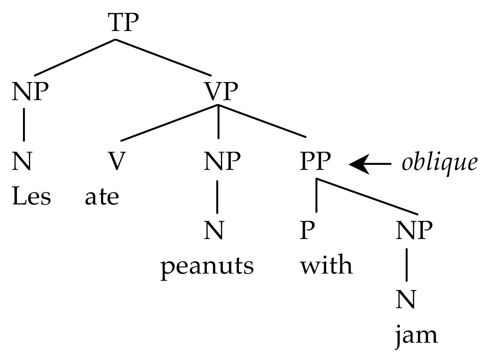
65) John tagged Lewis [$_{PP}$ *with a regulation baseball*][$_{PP}$ *on Tuesday*].

In many languages, such as Finnish, obliques aren't marked with prepositions, instead they get special suffixes that mark them as oblique; so obliqueness is not necessarily defined by being a preposition, that is just a convenient definition for now. 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 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.

66) a)



b)



You now have enough information to try General Problem Sets 10–16 and Challenge Problem Set 3

---

6. Summary and Conclusions

This chapter has been a bit different from the rest of this book. It hasn't been about Language per se, but rather about the mathematical properties of the system we use to describe language. We looked at the various parts of a syntactic tree and then at the three relations that can hold between these parts: domination, precedence, and c-command. In all the subsequent chapters of this book, you'll find much utility for the terms and the relations described here.

Ideas, Rules, and Constraints Introduced in this Chapter

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

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

iii) ***Label***: The name given to a node (e.g., N, NP, TP, etc.).

iv) ***(Proper) 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 branch from A to B going only downwards.

v) ***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.)

vi) A is the ***mother*** of B if A immediately dominates B.

vii) B is the ***daughter*** of A if B is immediately dominated by A.

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

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

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

xi) ***Non-terminal Node*** *(revised)*: A node that dominates something. (A node that is a mother.)

xii) ***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.

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

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

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

xvi) ***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.

xvii) ***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.

xviii) ***No Crossing Branches Constraint***: If node X precedes another node Y then X and all nodes dominated by X must precede Y and all nodes dominated by Y.

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

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

xxi) ***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.

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

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

xxiv) ***Government***: Node A governs node B if A c-commands B, and there is no node G, where G is c-commanded by A and G c-commands B.
- ***Phrase-government***: If A is a phrase, then the categories that count for G in the above definition must also be phrases
- ***Head-government***: If A is a head (word), then the categories that count for G in the above definition must also be heads.

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

xxvi) ***Object of Preposition*** *(preliminary)*: NP daughter of PP.

xxvii) ***Direct Object***:
a) With verbs of type $V_{[NP\_NP]}$, $V_{[NP\_ CP]}$ and $V_{[NP\_ NP\ PP]}$, the NP or CP daughter of VP.
b) With verbs of type $V_{[NP\_ NP\ \{NP/CP\}]}$, An NP or CP daughter of VP that is preceded by another NP daughter of VP (i.e., the second NP daughter of VP).

xxviii) ***Indirect Object*** *(preliminary):*
a) With verbs of type $V_{[NP\_ NP\ PP]}$, the PP daughter of VP immediately preceded by an NP daughter of VP.

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

xxix) ***Oblique:*** any NP/PP in the sentence that is not a subject, object of a preposition, or indirect object.

FURTHER READING

Barker, Chris and Geoffrey Pullum (1990) A theory of command relations. *Linguistics and Philosophy* 13, 1–34.

Chomsky, Noam (1975) *The Logical Structure of Linguistic Theory*. New York: Plenum.

Higginbotham, James (1985) A note on phrase markers. *MIT Working Papers in Linguistics* 6, 87–101.

Reinhart, Tanya (1976) The Syntactic Domain of Anaphora. Ph.D. dissertation. MIT.

Reinhart, Tanya (1983) *Anaphora and Semantic Interpretation.* London: Croom Helm.

GENERAL PROBLEM SETS

1. TREES

[Application of Skills; Basic to Intermediate]

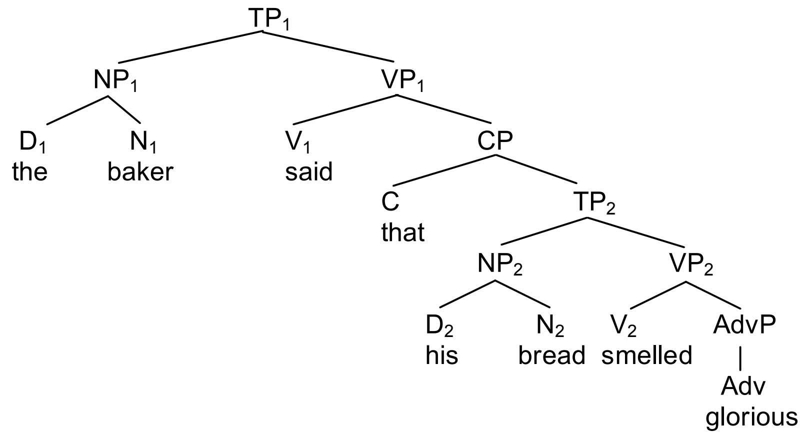
Using the rules we developed in chapter 3, draw the trees for the following sentences. Many of the sentences are ambiguous. For those sentences draw *one* possible tree, indicating the meaning by providing a paraphrase.

a) The big man from New York loves bagels with cream cheese.
b) Susan rode a bright blue train from New York.
c) The plucky platypus kicked a can of soup from New York to Tucson.
d) John said Martha sang the aria with gusto.
e) Martha said John sang the aria from *La Bohème*.
f) The book of poems with the bright red cover stinks.
g) Louis hinted Mary stole the purse deftly.
h) The extremely tired students hated syntactic trees with a passion.
i) Many soldiers have claimed bottled water quenches thirst best.
j) Networking helps you grow your business.

2. DOMINATION

[Application of Skills; Basic]

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



1) List all the nodes that dominate D_1 *the*.
2) List all the nodes that dominate D_2 *his*.
3) List all the nodes that dominate N_1 *baker*.
4) List all the nodes that dominate N_2 *bread*.
5) List all the nodes that dominate V_1 *said*.
6) List all the nodes that dominate V_2 *smelled*.
7) List all the nodes that dominate Adv *glorious*.
8) List all the nodes that dominate C *that*.
9) List all the nodes that dominate TP_1 (if there are any).
10) List all the nodes that dominate TP_2.
11) List all the nodes that dominate NP_1.
12) List all the nodes that dominate NP_2.
13) List all the nodes that dominate VP_1.
14) List all the nodes that dominate VP_2.
15) List all the nodes that dominate CP.
16) List all the nodes that dominate AdvP.
17) What is the root node?
18) List all the terminal nodes.
19) List all the non-terminal nodes.
20) List all the nodes that VP_2 dominates.
21) List all the nodes that CP dominates.
22) List all the nodes that NP_1 dominates.

3. EXHAUSTIVE DOMINATION

[Application of Skills; Intermediate]

Refer back to the tree for problem set 2 to answer this question.

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? If so, which one?
3) In the tree, is the set {V_2, Adv} exhaustively dominated by a single node? If so, which one?
4) In the tree, is the set {D_2, N_2, V_2, Adv} exhaustively dominated by a single node? If so, which one?
5) In the tree, is the set {D_1, N_1, V_1} exhaustively dominated by a single node? If so, which one?
6) In the tree, is the set {D_1} exhaustively dominated by a single node? If so, which one?
7) In the tree, is the set {C, D_2, N_2, V_2, Adv} exhaustively dominated by a single node? If so, which one?
8) What is the set of terminal nodes exhaustively dominated by VP_1?
9) Is the string *that his bread* a constituent? Explain your answer using the terminology of exhaustive domination.
10) Is the string *The baker said that his bread smelled glorious* a constituent? Explain your answer using the terminology of exhaustive domination.
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?

4. IMMEDIATE DOMINATION
[Application of Skills; Basic]
Go back to problem set 2, study the tree again and answer the questions (1–16) as in problem set 2, ***except*** limiting your answer to immediate domination instead of domination.

5. PRECEDENCE
[Application of Skills; Basic]
Go back to problem set 2, study the tree again and answer the questions (1–16) ***except*** changing domination to precedence (i.e., list all the nodes that precede D_1 etc.). For some elements there may be nothing that precedes them.

6. IMMEDIATE PRECEDENCE
[Application of Skills; Basic]
Go back to problem set 2, study the tree again and answer the questions (1–16) ***except*** changing domination to immediate precedence (i.e. list all the nodes that immediately precede D_1, etc.). For some elements there may be nothing that immediately precedes them.

7. C-COMMAND

[Application of Skills; Basic]

Go back to problem set 2, study the tree again and answer the following questions:

1) List all the nodes that D_1 *the* c-commands (note NOT the nodes that c-command D_1, but the ones that D_1 c-commands).
2) List all the nodes that D_2 *his* c-commands.
3) List all the nodes that N_1 *baker* c-commands.
4) List all the nodes that N_2 *bread* c-commands.
5) List all the nodes that V_1 *said* c-commands.
6) List all the nodes that V_2 *smelled* c-commands.
7) List all the nodes that Adv *glorious* c-commands.
8) List all the nodes that C *that* c-commands.
9) List all the nodes that TP_1 c-commands (if there are any).
10) List all the nodes that TP_2 c-commands.
11) List all the nodes that NP_1 c-commands.
12) List all the nodes that NP_2 c-commands.
13) List all the nodes that VP_1 c-commands.
14) List all the nodes that VP_2 c-commands.
15) List all the nodes that CP c-commands.
16) List all the nodes that AdvP c-commands.
17) What nodes c-command TP_2?
18) What nodes c-command NP_1?
19) What nodes c-command C?

8. SYMMETRIC AND ASYMMETRIC C-COMMAND

[Application of Skills; Basic/Intermediate]

Go back to problem set 2, study the tree again and answer the following questions. For some questions the answer may be "none":

1) List all the nodes that D_1 *the* symmetrically c-commands.
2) List all the nodes that D_2 *his* symmetrically c-commands.
3) List all the nodes that N_1 *baker* symmetrically c-commands.
4) List all the nodes that N_2 *bread* symmetrically c-commands.
5) List all the nodes that V_1 *said* symmetrically c-commands.
6) List all the nodes that V_2 *smelled* symmetrically c-commands.
7) List all the nodes that Adv *glorious* symmetrically c-commands.
8) List all the nodes that C *that* symmetrically c-commands.
9) List all the nodes that TP_1 symmetrically c-commands (if there are any).
10) List all the nodes that TP_2 symmetrically c-commands.
11) List all the nodes that NP_1 symmetrically c-commands.
12) List all the nodes that NP_2 symmetrically c-commands.
13) List all the nodes that VP_1 symmetrically c-commands.
14) List all the nodes that VP_2 symmetrically c-commands.
15) List all the nodes that CP symmetrically c-commands.
16) List all the nodes that AdvP symmetrically c-commands.

17) List all the nodes that D_1 *the* asymmetrically c-commands.
18) List all the nodes that D_2 *his* asymmetrically c-commands.
19) List all the nodes that N_1 *baker* asymmetrically c-commands.
20) List all the nodes that N_2 *bread* asymmetrically c-commands.
21) List all the nodes that V_1 *said* asymmetrically c-commands.
22) List all the nodes that V_2 *smelled* asymmetrically c-commands.
23) List all the nodes that Adv *glorious* asymmetrically c-commands.
24) List all the nodes that C *that* asymmetrically c-commands.
25) List all the nodes that TP_1 asymmetrically c-commands (if there are any).
26) List all the nodes that TP_2 asymmetrically c-commands.
27) List all the nodes that NP_1 asymmetrically c-commands.
28) List all the nodes that NP_2 asymmetrically c-commands.
29) List all the nodes that VP_1 asymmetrically c-commands.
30) List all the nodes that VP_2 asymmetrically c-commands.
31) List all the nodes that CP asymmetrically c-commands.
32) List all the nodes that AdvP asymmetrically c-commands.
33) What nodes asymmetrically c-command V_2?
34) What nodes symmetrically c-command NP_1?
35) What nodes asymmetrically c-command C?
36) What nodes symmetrically c-command C?

9. Government

[Application of Skills; Intermediate/Advanced]

Go back to problem set 2, study the tree again 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) What node(s) does C *that* head-govern?
6) Does NP_1 phrase-govern AdvP? Why or why not?
7) Does VP_2 phrase-govern N_2? Why or why not?

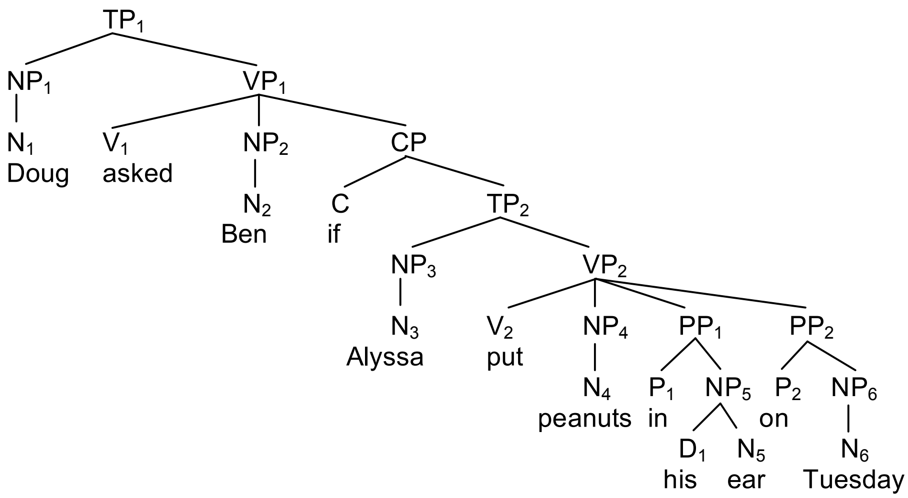
10. Grammatical Relations I

[Application of Skills; Intermediate]

Go back to problem set 2, study the tree again and answer the following questions:

1) What is the subject of TP1?
2) What is the subject of TP2?
3) What is the object of VP1?
4) Does VP2 have an object? Why or why not?

11. GRAMMATICAL RELATIONS II
[Application of Skills; Intermediate]
Examine the following tree and then answer the questions that follow:



1) What is the subject of TP_1?
2) What is the subject of TP_2?
3) What is the object of P_1?
4) What is the object of P_2?
5) What is the direct object of VP_1?
6) What is the direct object of VP_2?
7) What is the indirect object of VP_1?
8) What is the indirect object of VP_2?
9) Is PP_2 an indirect object or an oblique. How can you tell?

12. GRAMMATICAL RELATIONS III[8]
[Application of Skills and Data Analysis; Basic]
For each of the following sentences, identify the subject, the object (if there is one), the indirect object (if there is one), any objects of prepositions, the verb, and any obliques. Draw the tree for each sentence.

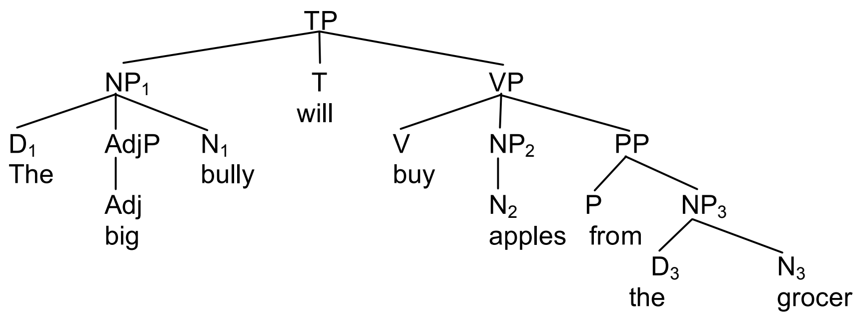
a) It never rains violently in southern California.
b) Soon we should give the family dog another bath.
c) The quiz show contestant bravely made a wild guess about the answer.

[8] Problem set contributed by Sheila Dooley Collberg.

13. STRUCTURAL RELATIONS I[9]

[Application of Skills; Advanced]

Consider the following tree:



1) What node(s) dominate N_3 *grocer*?
2) What node(s) immediately dominate D_3 *the*?
3) Do T *will* and V *buy* form a constituent?
4) What nodes does N_1 *bully* c-command?
5) What nodes does NP_1 *the big bully* c-command?
6) What is V *buy*'s mother?
7) What nodes does T *will* precede?
8) List all the sets of sisters in the tree.
9) What is the PP's mother?
10) Do NP_1 and VP asymmetrically or symmetrically c-command one another?
11) List all the nodes c-commanded by V.
12) What is the subject of the sentence?
13) What is the object of the sentence?
14) What is the object of the preposition?
15) Is NP_3 a constituent of VP?
16) What node(s) is NP_3 an immediate constituent of?
17) What node(s) does VP exhaustively dominate?
18) What is the root node?
19) List all the terminal nodes.
20) What immediately precedes N_3 *grocer*?

14. STRUCTURAL RELATIONS II

[Application of Skills; Advanced]

Look at your tree for sentence (a) of problem set 1. Number the nodes the way I did for problem set 13.

1) List all the nodes that the subject NP c-commands.
2) List all the nodes that the subject NP asymmetrically c-commands.
3) List all the nodes that the subject NP dominates.

---

[9] The idea for this problem set is borrowed from Radford (1988).

4) List all the nodes that the subject NP immediately dominates.
5) List all the nodes that the subject NP precedes.
6) List all the nodes that the VP node c-commands.
7) List all the nodes that the VP asymmetrically c-commands.
8) List all the nodes that the VP dominates.
9) List all the nodes that the VP immediately dominates.
10) List all the nodes that the VP precedes.
11) List all the nodes that the VP follows (i.e., is preceded by).

15. TZOTZIL

[Data Analysis; Basic]

Tzotzil is a Mayan language spoken in Mexico. Consider the following sentences, then answer the questions that follow. Glosses have been simplified and the orthography altered from the original source. (Data from Aissen 1987.)

a) 'ispet lok'el 'antz ti t'ule.
carry away woman the rabbit
"The rabbit carried away (the) woman."

b) 'ibat xchi'uk smalal li Maruche.
go with her-husband the Maruch
"(the) Maruch went with her husband." (Maruch is a proper name.)

c) Pas ti 'eklixa'une.
built the church
"The church was built."

1) What is the NP rule for Tzotzil?
2) What is the PP rule for Tzotzil?
3) What is the VP rule for Tzotzil?
4) What is the TP rule for Tzotzil?
5) What is the subject of sentence (b)?
6) Is *[the church*] a subject or an object of sentence (c)?
7) Does the verb precede the subject in Tzotzil?
8) Does the object precede the subject in Tzotzil?
9) Does the verb precede the object in Tzotzil?
10) Using the rules you developed in (1–4) above, draw the trees for (b) and (c).

16. HIAKI

[Data Analysis; Intermediate]

Consider the data from the following sentences of Hiaki (also known as Yaqui), an Uto-Aztecan language from Arizona and Mexico. Data have been simplified. (Data from Dedrick and Casad 1999.)

a) Tékil né-u 'aáyu-k.
work me-for is
"There is work for me." (literally: "Work is for me.")

b) Hunáa'a yá'uraa hunáka'a hámutta nokriak.
   that chief that woman defend
   "That chief defended that woman."

c) Taáwe tótoi'asó'olam káamomólim híba-tu'ure.
   Hawk chickens young like
   "(The) hawk likes young chickens."

d) Tá'abwikasu 'áma yépsak.
   different-person there arrived
   "A different person arrived there." (assume *there* is an adverb not a N)

Assume the rules AdjP → Adj and AdvP → Adj and answer the following questions.

1) What is the NP rule for Hiaki?
2) Do you need a PP rule for Hiaki? Why or why not?
3) What is the VP rule for Hiaki?
4) What is the TP rule for Hiaki?
5) Using the rules you developed in questions 1–4, draw the tree for sentences (b, c, d).
6) What is the subject of sentence (b)?
7) Is there an object in (d)? If so, what is it?
8) What node(s) does *hunáa'a* c-command in (b)?
9) What node(s) does *hunáa'a yá'uraa* c-command in (b)?
10) What does '*áma* precede in (d)?
11) What node immediately dominates *káamomólim* in (c)?
12) What nodes dominate *káamomólim* in (c)?
13) What node immediately precedes *káamomólim* in (c)?
14) What nodes precede *káamomólim* in (c)?
15) Does *káamomólim* c-command *táawe* in (c)?
16) Do *hunáka'a* and *hámutta* symmetrically c-command one another in (b)?

CHALLENGE PROBLEM SETS

CHALLENGE PROBLEM SET 1: DISCONTINUOUS CONSTITUENTS
[Critical Thinking; Challenge]
Consider the following data:

a) A woman entered who was eating a chocolate enchiladas.
b) The man that Bill said that Mary disliked loves beef waffles.

With sentence (a) assume that the relative clause *[who was wearing a hat]* is a modifier of *man*. Assume that *the man* is both the direct object of the verb *disliked* and the subject of the verb *loves*. Is it possible to draw trees for these sentences without crossing lines? Explain why or why not.

CHALLENGE PROBLEM SET 2: NEGATIVE POLARITY ITEMS

[Critical Thinking; Challenge]

There is a class of phrase, such as [a red cent] and [a single thing], that are called Negative Polarity Items (NPI). These are only allowed in sentences with a negative word like *not*. So for example, in sentences (a) and (c) the NPI is fine, in the (b) and (d) sentences, however, the sentence is at best strange.

a) I didn't have a red cent.
b) *I had a red cent. *(ungrammatical with idiomatic reading)*
c) I didn't read a single book the whole time I was in the library.
d) *I read a single book the whole time I was in the library.

It turns out that sentences with NPIs not only must have a word like *not*, they also have to be in a particular structural relationship with that *not* word. On the basis of the following sentences figure out what that relationship is. There are two possible answers consistent with this data. Assume that *not* and *n't* are dominated by the VP node.

e) I did not have a red cent.
f) *A red cent was not found in the box.

What kind of data would you need to decide between the two possible answers to this question?

CHALLENGE PROBLEM SET 3: IRISH

[Data Analysis and Critical Thinking: Challenge]

Consider the following data from Modern Irish Gaelic:

a) Phóg Liam Seán.
kissed William John
"William kissed John."

b) Phóg Seán Liam.
Kissed John William
"John kissed William."

c) Phóg an fear an mhuc.
kissed the man the pig
"The man kissed the pig."

d) Chonaic mé an mhuc mhór.
Saw I the pig big
"I saw the big pig."

e) Rince an bhean.
Danced the woman
"The woman danced."

On the basis of this data answer the following questions:

1) What is the AdjP rule in Irish (if there is one)? Constrain your answer to the data here.
2) Write the NP rule for Irish, be sure to mark optional things in parentheses.
3) Can you write a VP rule for Irish? Assume that if you have a VP then object NPs (like *William* in (b) and *the big pig* in (d)) *must* be part of the VP, and that subject NPs (like *John* in (b) and *I* in (d)) are *never* part of VPs. Is it possible to keep those assumptions and not cross lines? ***If you can't, then don't posit a VP***.
4) If you don't have a VP rule for Irish, then how do we define direct object in this language?
5) What is the TP rule for Irish? (Be careful that your TP rule is consistent with your answer in (3).)
6) Using the rules you developed, draw trees for sentences (c), (d) and (e).