CAS LX 522
Syntax I

Week 2. Clauses and Trees
and c-command, oh my.

Previously, in LX 522…

Sentences have structure, and are made up of constituents.
The constituents are phrases.
A phrase consists of a head and modifiers.
The categorial type of the head determines the categorial type of the phrase (e.g., a phrase headed by a noun is a noun phrase).
There are several categories, we looked at some of them and determined phrase structure rules or templates for what each kind of phrase can contain.

Previously, in LX 522…

- We looked at NP, VP, PP, AdvP, and AdjP.
  - NP: (D) (AdjP+) N (PP+)
  - PP: P (NP)
  - VP: (AdvP+) V (NP) (PP+) (AdvP+)
  - AdjP: (AdvP) Adj
  - AdvP: (AdvP) Adv

Trees and constituency

- A sentence has a hierarchical structure
- Constituents can have constituents of their own.
- The simplest way to draw this is with a tree.

Trees

- The tree diagram is the most important analytical notation we will work with, and we will use a lot of trees through the semester, so it is important to be able to understand and draw trees.

Step 1: Identify categories

The first step is to identify the category of each of the words in the sentence.
Step 2: Locate modification

The second step is to figure out the modification relations between words. What modifies what? Here, we have several *thes* and each modifies the noun to its right.

```
D N V D N P D N
The student put the book on the table
```

Step 3: Apply rules

The third step is to apply our rules, remembering the Golden Rule of Modifiers: *Modifiers are always attached within the phrase they modify.* So we look at the things being modified, and consult the rule for things of that category.

```
D N V D N P D N
The student put the book on the table
```

Step 3: Apply rules

We have several N’s being modified. So we consult our rule about NPs:
- NP: (D) (AdjP+) N (PP+)
Starting at the right edge, we see that *the table* can form an NP.

```
D N V D N P D N
The student put the book on the table
```

Step 3: Apply rules

So, we draw an NP above *the table.* Now, consider *on.* It is a P, and there is only one kind of phrase which can contain a P:
- PP: P NP
Can we build a PP with what we have?

```
D N V D N P D N
The student put the book on the table
```

Step 3: Apply rules

Sure, we can draw in a PP for *on the table.* Next, look at *book.* It is an N and the only rule we have that contains an N is NP:
- NP: (D) (AdjP+) N (PP+)
Can we build an NP?

```
D N V D N P D N
The student put the book on the table
```

Step 3: Apply rules

Here, we have two choices.
- NP: (D) (AdjP+) N (PP+)
An NP may but need not contain a PP. We have D N PP at our disposal. We could put them all in an NP, or we could leave the PP out of the NP.

```
D N V D N P D N
The student put the book on the table
```
Step 3: Apply rules

Only one choice is the right choice. How do we know which one it is?
Answer: The Golden Rule of Modifiers.

Step 3: Apply rules

In *The student put the book on the table*, does *on the table* modify *book*? If so, it needs to be in the NP headed by *book*.

Step 3: Apply rules

Compare this sentence to

*The student saw the book on the table*

What is the difference them with respect to *on the table*?

Step 3: Apply rules

*On the table* in our sentence modifies *put* (it specifies the goal location of the putting); it does not modify *book*, and so it should not be included in the same NP as *book* (it should be in the same phrase as *put*).

Step 3: Apply rules

The last step we can do with the rules we have so far is to build the NP over *the student*.
Step 3: Apply rules
Using the idea that the sentence has an NP and a VP (which we will soon add to our rules), we can complete the tree.

The student put the book on the table

Step 3: Apply rules
And that’s our tree for The student put the book on the table.

The S node
- At the end of our tree, we had to posit a rule which we hadn’t yet formalized:
  - S: NP VP
- This is a good first approximation, but there are a couple of problems with this formulation

The S node
- The first problem is that it is not complete as it stands. Consider:
  - The students will eat the sandwiches.
  - We have an NP the students, which is the subject of the sentence.
  - We have an NP the sandwiches and a VP eat the sandwiches.
  - But what is will?

The S node
- There are a number of things which can go in this position. One group of these are called modals:
  - Pat could leave.
  - Pat should leave.
  - Pat might leave.
  - Pat will leave.
  - Pat would leave.
- Modals appear between the subject NP (Pat) and the VP (leave). So, we need to allow for this in our S rule.

The S node
- We also need to allow for the not in negative sentences like:
  - Pat might not leave.
  - Pat should not leave.
- So, we now have
  - S: NP (Modal) (Neg) VP
Do-support

- Pat left.
- Pat did not leave.
- "Pat not left.
- When you negate a sentence like this in English, you need to use do.
- Do looks like it is in the same place that modals are.
- When you use do like this, do gets marked for tense, not the verb.

Do-support

- In fact, when you have something in the “Modal” slot, the verb never shows past tense marking.
- Pat left.
- Pat will (not) leave.
- Pat did not leave.
- Pat should not leave.
- Hypothesis: The “modal” slot is where the tense marking (past, present, future) goes.

Do-support

- For this reason, we will call the “modal” slot “T” (for “tense”).
- S: NP (T) (Neg) VP

Terminology note: This position has gone by several names in linguistic history.
- 60’s: Aux (auxiliary, for have and be)
- 80’s: Infl or I (inflection, including tense)
- 90’s: Tns or T (sort of, to be clarified later)

Do-support

- Now, consider Pat left. The verb is marked with past tense, but we wanted to make T be where the tense information goes.
- The common view is that T holds something that is smaller than a word, a tense affix.

The tense affix

- If you look at verbs, many of them can be distinguished in the present and the past tense by the presence of -ed at the end.
  - Walk vs. walked (walk+ed)
  - Wait vs. waited (wait+ed)
  - Sleep vs. slept (sleep+ed)
- The idea is that the past tense of the verb is made of the verb stem plus something else, the past tense suffix.

The tense affix

- If we suppose that the past tense affix -ed is of category T, we could write Pat left this way:
  - Pat -ed leave
- Part of being a verbal affix (in this case a verbal suffix) is that it is required to be attached to a verb.
- So -ed must “hop” onto leave (because verbal affixes need to be attached to verbs), yielding left.
The tense affix

- Now, since every sentence needs tense, we can suppose that the T in our S rule isn’t optional—there is always a T there, but it can be an affix which will hop onto the verb and be pronounced as one word with the verb.
- S: NP T (Neg) VP

Do-support

- This also gives us an explanation for why when you negate a sentence you need to use do:
  - Pat did not leave.
- The past tense affix needs to be attached to a verb, but it can’t because not is in the way.
- The way out is to insert a “dummy verb”, a verb that has no semantic content, that -ed can attach to.

Do-support

- The idea is that we insert the “dummy verb” do as a “last resort” if the sentence has a “stranded affix” that can’t hop onto an adjacent verb. This is called do-support.

The S node

- So given “affix hopping” and do-support, we can write our S rule with three required elements:
  - S: NP T (Neg) VP
- There is something else which is unusual about the S rule in comparison to our other rules.

The S node

- Compare S: NP T (Neg) VP to
  - NP: (D) (AdjP+) N (PP+)
  - PP: P (NP)
  - VP: (AdvP+) V (NP) (PP+) (AdvP+)
- Our other rules make phrases that are the same category as their head.
- What is the head of S?

The TP node

- An obvious choice, now that T is a required part of S, is to assume that T is the head of S.
- Given this, we will rename our S node to “TP” to be more in line with our other phrases.
- TP: NP T (Neg) VP
- That is, the tense morpheme -ed or a modal like might is actually the head of the sentence.
- Terminology reminder: In 80's terminology this is called “IP” and is often referred to that way even today—it's the same thing.
Embedded clauses

- There is just one more kind of phrase we should add to our set of structure rules.
- It is possible to embed one sentence inside another, like this:
  - Pat said that the students ate the sandwiches.
- The whole thing is a sentence, but it has our familiar sentences as part of it.

The CP

- Pat said that TP.
- We can write a rule for CP like this, where *that* (C) is the head, and TP is an obligatory “modifier.”
  - CP: C TP
- And we need to modify our VP rule to allow CP to be the object of a verb (like *say*):
  - VP: (AdvP+) V ((NP/CP)) (PP+) (AdvP+)
- Rarely, you’ll see CP referred to as S’ (“S-bar”); that is 70s terminology. If you see S’, think CP.

Our phrase structure rules

- We now have a fairly complete set of rules.
  - NP: (D) (AdjP+) N (PP+)
  - PP: P (NP)
  - VP: (AdvP+) V ((NP/CP)) (PP+) (AdvP+)
  - AdjP: (AdvP) Adj
  - AdvP: (AdvP) Adv
  - TP: [NP/CP] T (Neg) VP
  - CP: C TP

Embedded clauses

- Pat said that the students ate the sandwiches.
- We know that *the students ate the sandwiches* is a TP, so let’s abbreviate this:
  - Pat said that TP.
- When you embed a sentence, you generally need a word like *that*, called a complementizer. We will assign it to category C.

The CP

- In fact, a CP can not only be the object of a verb, but it can also be the subject of a verb:
  - That Pat left surprised me.
  - The dog surprised me.
- So, we need to allow for this in our TP rule:
  - TP: [NP/CP] T (Neg) VP

Recursion

- An important property of the rules we have is that they are recursive. Inside a CP, you can have a CP. Inside an AdvP you can have an AdvP. This means that there in principle an infinite number of possible sentence structures.
  - John left.
  - Mary said that John left.
  - Bill thinks that Mary said that John left.
  - I heard that Bill thinks that Mary said that John left.
  - Pat said that I heard that Bill thinks that Mary said that John left.
Back to the trees

- We now have the tools to draw trees for a lot of English sentences. Let’s do another one—it will be very important to be comfortable with converting sentences into trees.
- Our sentence will be:
  - John said that the dog barked very loudly.

Step 1: Identify categories

- First, identify the categories.
- John said that the dog barked very loudly.

Step 2: Locate modification

- First, identify the categories.
- Then, figure out what modifies what.
- The modifies dog.
- Very modifies loudly.
- Very loudly modifies barked.
- Now, we start to apply our rules.
- John said that the dog barked very loudly.

Step 3: Apply rules

- Very modifies loudly, so very must be inside the phrase headed by loudly, an AdvP. Our rule is:
- Notice: The AdvP headed by loudly can optionally take an AdvP—not an Adv. So, first we need to make very an AdvP.
- John said that the dog barked very loudly.
Step 3: Apply rules

- Next, we have the V. Our rule is:
  - VP: (AdvP+) V (NP/CP) (PP+) (AdvP+)
- So we can build a VP containing the verb and the AdvP very loudly.

Step 3: Apply rules

- Moving on to dog, it is modified by the, together constituting the subject NP of the embedded sentence. Our rule allows us to build an NP here.
  - NP: (D) (AdjP+) N (PP+)
Step 3: Apply rules

- Next, the VP rule to combine say and the CP.
  \[\text{VP: (AdvP+)} \ V \ ((\text{NP/CP}) \ (\text{PP+}) \ (\text{AdvP+})\]

\[
\begin{array}{c}
\text{Step 3: Apply rules}\\
\end{array}
\]

- And then the TP rule: TP: {NP/CP} T (Neg) VP.
  - This needs an NP, so we need to build that first.

\[
\begin{array}{c}
\text{Step 3: Apply rules}\\
\end{array}
\]

That vs. Ø

- What if the sentence were
  - John said the dog barked very loudly?

\[
\begin{array}{c}
\text{That vs. Ø}\\
\end{array}
\]

One to try...

- NP: (D) (AdjP+) N (PP+)
- PP: P (NP)
- VP: (AdvP+) V ((NP/CP)) (PP+) (AdvP+)
- AdjP: (AdvP) Adj
- AdvP: (AdvP) Adv
- TP: [NP/CP] T (Neg) VP
- CP: C TP

- The young consumers walked to the new store.
Trees

- The “joints” of the tree are nodes. The nodes here are labeled (with node labels).
- Nodes are connected by branches.
- The node at the top of the tree (with no branches above it) is called the root node. A is the root node.

Nodes with no branches beneath them are called terminal nodes.
- B, D, E, F are terminal nodes.

Nodes with branches beneath them are called nonterminal nodes.
- A, C are nonterminal nodes.

Tree relations

- A node X dominates nodes below it on the tree; these are the nodes which would be pulled along if you grabbed the node X and pulled it off of the page.
  - C dominates D, E, and F.

A node X immediately dominates a node Y if X dominates Y and is connected by only one branch.
- A immediately dominates B and C.
- A is also sometimes called the mother of B and C.

A node which shares the same mother as a node X is sometimes called the sister of X.
- B is the sister of C.
- C is the sister of B.
- D, E are the sisters of F.

A node X ccommands its sisters and the nodes dominated by its sisters.
- B ccommands C, D, E, and F.
- D ccommands E and F.
- C ccommands B.

C-command is very important to understand!
Tree relations

- What does PP dominate?

Tree relations

- What is/are the sister(s) of V?

Tree relations

- What is/are the sister(s) of the N store?

Tree relations

- What does P c-command?

Tree relations

- What does VP c-command?

Grammatical relations

- What is the subject of this sentence?
  - The NP The young consumers.
- Notice that this is the daughter of TP.
Grammatical relations

- In fact, the subject is in general, the NP which is the daughter of TP.
  - **Subject** = NP daughter of TP

Grammatical relations

- Similarly the (direct) object is generally the NP which is the daughter of VP
  - **Direct object** = NP daughter of VP.

Grammatical relations

- And the object of a preposition is the NP which is the daughter of PP.
  - **Object of a preposition** = NP daughter of PP.

Precedence

- The tree also encodes the linear order of the terminal nodes.
  - The is pronounced before students.
  - Saw is pronounced before the and students.

Precedence

- That is, V is pronounced before NP, meaning V is pronounced before all of the terminal nodes dominated by NP.

Precedence

- Even if the tree is drawn sloppily, nothing changes—(everything dominated by) V is pronounced before (everything dominated by) NP. This is still “saw the students”.
No line crossing

- One of the implications of this is that you cannot draw a well-formed tree with lines that cross.
- Adv can't be pronounced before V because Adv is part of NP and V has to be pronounced before all of NP.

Back to c-command

- To reiterate, c-command is a very important concept of tree geometry. It’s not fundamentally complicated, but it turns out to be very useful in characterizing natural language syntax.
- A node X c-commands its sisters and the nodes dominated by its sisters.

Back to c-command

- C-command can be symmetric or asymmetric.
- A node X asymmetrically c-commands a node Y if X c-commands Y but Y doesn’t c-command X.
- Two nodes X and Y symmetrically c-command each other if X c-commands Y and Y c-commands X.
- Examples?

Bracketed diagrams

- Another way of conveying the information in a tree is with a bracketed diagram.
- [\[A B \[C D E F\]\]]
- Constituents are marked by matching brackets, the left bracket (generally) is marked with the label of the constituent as a whole.

Negative Polarity Items

- Certain words in English seem to only be available in “negative” contexts.
- Pat didn’t invite anyone to the party.
- Pat does not know anything about syntax.
- Pat hasn’t ever been to London.
- Pat hasn’t seen Forrest Gump yet.
- *Pat invited anyone to the party.
- *Pat knows anything about syntax.
- *Pat has ever been to London.
- *Pat has seen Forrest Gump yet.

Negative Polarity Items

- These are called negative polarity items.
- They include ever, yet, anyone, anything, any N, as well as some idiomatic ones like lift a finger and a red cent.
- *Pat didn’t lift a finger to help.
- *Pat didn’t have a red cent.
- *Pat lifted a finger to help.
- *Pat had a red cent.
Any

- Just to introduce a complication right away, there is a positive-polarity version of *any* that has a different meaning, known as the “free choice *any*” meaning. This meaning is distinguishable (intuitively) from the NPI *any* meaning, and we are concentrating only on the NPI *any* meaning—for now, we will just consider *any* to be ambiguous, like *bank*.
  - John read anything the professor gave him.
  - Anyone who can understand syntax is a genius.
  - Pick any card.

Licensing

- NPI’s are only allowed to appear if there’s a negative in the sentence.
  - We say that negation gives them “license to appear”: NPI’s are licensed by negation in a sentence.
    - Just like you need a driver’s license to drive a car (legally), you need negation to use a NPI (grammatically).

Negative Polarity Items

- But it isn’t quite as simple as that. Consider:
  - I didn’t see anyone.
  - *I saw anyone.
  - *Anyone didn’t see me.
  - *Anyone saw me.
  - It seems that simply having negation in the sentence isn’t by itself enough to license the use of an NPI.

Negative Polarity Items

- As a first pass, we might say that negation has to precede the NPI.
  - I didn’t see anyone.
  - *Anyone didn’t see me.
  - But that’s not quite it either.
    - *(That John didn’t stay)* surprised anyone.
    - *(That John didn’t stay)* didn’t surprise anyone.

Negative Polarity Items

- *(That John didn’t stay)* surprised anyone.
- *(That John didn’t stay)* didn’t surprise anyone.

Structural ambiguity

- John said that Bill slipped in the kitchen.
  - This sentence has two possible meanings; either John said it in the kitchen, or Bill slipped in the kitchen (according to John).
  - John said that Bill will leave yesterday.
  - John said that Bill will leave tomorrow.
**Structural ambiguity**

```
NP       TP
  John -ed
  V   say
  C that
  NP
  Bill -ed
  V   slip
  PP in the kitchen
```

**Negative Polarity Items**

- John said that Bill didn’t slip in any room in the house.
- Suddenly, it has only one meaning. Why?
  - John said: In no room did Bill slip.
  - *John said in any room: Bill didn’t slip.

**Negative Polarity Items**

- How about:
  - John didn’t say that Bill slipped in any room in the house.
  - What do we predict?

**Negative Polarity Items**

- John didn’t say that Mary slipped in any room in the house.
  - …He said that when he was out in the yard…
  - …He said that she slipped on the sidewalk…
- Both meanings are good, because both possible structural positions for the NPI are c-commanded (thus licensed) by the negation.
**Binding Theory**

- **Binding Theory** is primarily concerned with explaining the distribution of three kinds of noun phrases:
  - **Anaphors.** Expressions like *himself, herself, myself, each other.*
  - **Pronouns.** Expressions like *him, her.*
  - **R-expressions.** Referring expressions like *Pat, Chris.*

**R-expressions**

- R-expressions are NPs like *Pat, or the professor, or an unlucky farmer,* which get their meaning by referring to something in the world. Most NPs are like this.

**Anaphors**

- An anaphor does *not* get its meaning from something in the world—it depends on something else in the sentence.
  - John saw himself in the mirror.
  - Mary bought herself a sandwich.

**Pronouns**

- A pronoun is similar to an anaphor in that it doesn’t refer to something in the world but gets its reference from something else.
  - John told Mary that he likes pizza.
  - Mary wondered if she agreed.
  - Mary concluded that he was crazy.
  - …but it doesn’t *need* to be something in the sentence.

**Anaphors and pronouns**

- Anaphors and pronouns are *referentially dependent,* they do not have an intrinsic meaning.
  - Anaphors: *himself, herself, myself, yourself, itself, themselves, yourselves, ourselves.* Very similar are reciprocals like each other.
  - Pronouns: he, him, she, her, I, me, you, them, it, we, us.

**The problem**

- It turns out that there are very specific configurations in which pronouns, anaphors, and R-expressions can/must be used.
  - Even though both *he* and *himself* could refer to *John* below, you can’t just choose freely between them.
    - John saw himself.
    - “John saw him.
    - John thinks that Mary likes him.
    - “John thinks that Mary likes himself.
    - John thinks that he is a genius.
    - “John thinks that himself is a genius.
The problem

- The question Binding Theory strives to answer is:
- When do you use anaphors, pronouns, and R-expressions?

Indices and antecedents

- Anaphors and pronouns are referentially dependent; they can (or must) be co-referential with another NP in the sentence.
- The way we indicate that two NPs are co-referential is by means of an index, usually a subscripted letter. Two NPs that share the same index (that are coindexed) also share the same referent.
- John saw himself, in the mirror.

Indices and antecedents

- John saw himself, in the mirror.
- The NP from which an anaphor or pronoun draws its reference is called the antecedent.
- John is the antecedent for himself. John and himself are co-referential.

Constraints on co-reference

- John saw himself.
- Himself saw John.
- *John’s mother saw himself.
- It is impossible to assign the same referent to John and himself in the second and third sentences. What is different between the good and bad sentences?

Binding

- What is the difference between the relationship between John and himself in the first case and in the second case?

Binding

- In the first case, the NP John c-commands the NP himself. But not in the second case.
Binding

- When one NP c-commands and is coindexed with another NP, the first is said to bind the other.

Principle A

- Principle A of the Binding Theory (preliminary):
  An anaphor must be bound.

Binding domains

- But this is not the end of the story; consider
  - "John said that himself likes pizza.
  - "John said that Mary called himself.
- In these sentences the NP John c-commands and is coindexed with (=binds) himself, satisfying our preliminary version of Principle A—but the sentences are ungrammatical.
  - John didn’t say that anyone likes pizza.
  - John didn’t say that Mary called anyone.

Binding

- Definition: A binds B iff
  - A c-commands B
  - A is coindexed with B “if and only if”

Principle A

- This also explains why the following sentences are ungrammatical:
  - "Himself saw John in the mirror.
  - "Herself likes Mary’s father.
  - "Himself, likes Mary’s father.
- There is nothing which c-commands and is coindexed with himself and herself. The anaphors are not bound, which violates Principle A.

Binding domains

- John saw himself in the mirror.
- John gave a book to himself.
- "John said that himself is a genius.
- "John said that Mary dislikes himself.
- What is wrong? John binds himself in every case. What is different?
- In the ungrammatical cases, himself is in an embedded clause.
Binding domains

- It seems that not only does an anaphor need to be bound, it needs to be bound nearby (or locally).

- **Principle A (revised):**
  - An anaphor must be bound in its binding domain.

  **Binding Domain (preliminary):**
  - The binding domain of an anaphor is the smallest clause containing it.

Pronouns

- *John, saw him, in the mirror.
- John, said that he, is a genius.
- John, said that Mary dislikes him,.
- John, saw him, in the mirror.
- How does the distribution of pronouns differ from the distribution of anaphors?
- It looks like it is just the opposite.

Principle B

- **Principle B**
  - A pronoun must be free in its binding domain.

  **Free**
  - Not bound

  - *John, saw him,.
  - John,’s mother saw him,.

Principle C

- We now know where pronouns and anaphors are allowed. So what’s wrong with these sentences? The pronouns are unbound as needed for Principle B. What are the binding relations here?

  - *He, likes John,.
  - *She, said that Mary, fears clowns.
  - His, mother likes John,.

Principle C

- Binding is a means of assigning reference.
- R-expressions have intrinsic reference; they can’t be assigned their reference from somewhere else.
- R-expressions can’t be bound, at all.

  - **Principle C**
    - An r-expression must be free.

Binding Theory

- **Principle A**
  - An anaphor must be bound in its binding domain.

- **Principle B**
  - A pronoun must be free in its binding domain.

- **Principle C**
  - An r-expression must be free.

- In several weeks, we will return to the Binding Theory to revise the definition of binding domain (it is more complicated than “smallest clause”).