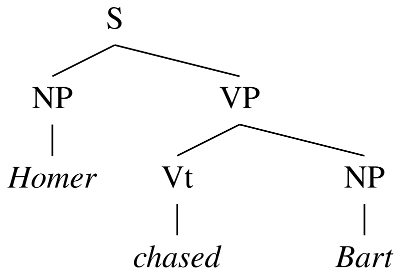
CAS LX 321 / GRS LX 621 *Syntax: Introduction to Sentential Structure*

September 25ish, 2018

1 Recapping/expanding some previous discussion

1.1 Tree relations



- **Dominance**. Node X dominates node Y if a downward path connects X to Y.
- **Precedence**. Node X precedes node Y if neither dominates the other and X is left of Y.
- **C-command**. Node X c-commands node Y if neither dominates the other and the first branching node Z that dominates X also dominates Y.

NOTE: The definition of c-command I described in class initially differs a little bit from how I'd actually written it on the preceding handout (and above). The difference has to do with what a node without siblings is understood to c-command. So, let me lay out a little bit more clearly here what this definition entails.

We are always considering two nodes when we are evaluating whether one c-commands the other or not. Whichever nodes we're picking, we'll call one X and the other Y. If X dominates Y, or if Y dominates X, then neither one c-commands the other. C-command and dominance are mutually exclusive relations. So, VP does not c-command *Bart* above, nor does *Bart* c-command VP.

Now that we know that neither X nor Y dominates the other, we move up the tree from X to the node above it. If the node above it dominates anything that doesn't dominate X (that is, if there is more than one branch down from it), we stop; that is a "branching node." We call it Z. So, if X is *chased* and we move up one node, we get to Vt. Vt does not dominate anything that doesn't dominate *chased*, so it is *not* a branching node. We proceed up to its mother node, VP. VP *is* a branching node, because it dominates both (Vt and) *chased* and NP and *Bart*. So, we call VP "Z."

The last step is to see if Z, the first branching node dominating X, also dominates Y. In the example above, Z is VP (the first branching node dominating *chased*, a.k.a. X). If Y is *Bart*, then Z does dominate it. Which, ultimately, means that *chased* c-commands *Bart*. (If Y were *Homer*, Z would not dominate it, and thus: *chased* does not c-command *Homer*.)

This is a bit more complicated than the version I promoted in class initially, in that it allows us to "discount" non-branching nodes. One is right and one isn't, given certain assumptions about how the structures are constructed, and the data it is supposed to account for (e.g., anaphors, NPIs), but in a lot of cases it doesn't make any difference. In the homework, for example, using the interpretation of c-command here on the handout led to the same results as using the interpretation I introduced previously in class.

Unless we find reason later to change this, however, we'll stick to the version here on the handout since that's the one you have available for reference.

1.2 Lexicon and subcategorization frames

Conceptually there are two different kinds of rewrite rules that we'd been using, those that determine the tree strucutre (like S $\rightarrow$ N VP) and those that determine how the lexical items fit into the tree (like N $\rightarrow$ *Homer*). It is going to be useful to treat these more differently. Specifically, to factor out the lexical items into a **Lexicon**.

A lexical item has the following structure:

word, category, features

The category is something like V, N, P, Det. The features are things like [+pl] for plural.

There is a particular kind of a feature that determines whether a lexical item will "fit" into the structure, which is called a **subcategorization frame**. This is what distinguishes intransitive, transitive, and ditransitive verbs, for example.

- [+ __] is the subcategorization frame for an instransitive verb.
- [+ __ NP] is the subcategorization frame for a transitive verb.
- [+ __ NP NP] is the subcategorization frame for a ditransitive verb.

This feature is specifying the *context* in which this lexical item is allowed to appear. In order to use this lexical item in the tree it must fit in the context. The __ represents the item itself. The [+] notation is basically saying "this is a feature." So, the ditransitive subcategorization frame above says that the verb must occur in a constituent that consists of the verb and two following NPs (e.g., *gave Lisa a book*).

If there is no subcategorization frame on a lexical item, it is taken to be unconstrained. If there are multiple subcategorization frames on a lexical item, then it is taken to be sufficient for at least one of them to be met.

For example:

donate, V, [+ __ NP PP]

So, *donate* is a V (a lexical item of the "verb" category) that must occur in a constituent constisting of itself, an NP, and a PP.

1.3 The lexicon vs. the phrase structure rules

We will make the following assumptions—which we didn't *have* to make, but just as a way to constrain our analyses—about the nature of the PSRs and lexical entries:

- PSRs are *context-free*, meaning that the cannot restrict the situations in which they apply. If you have a PSR that allows a PP to be rewritten, you can rewrite any PP there is. You can't say, for example, that a rule can rewrite a PP into something only if that PP follows a VP.
- A subcategorization frame (context-sensitivity) can be added to a lexical item, but it can only constrain the immediate constituent containing the lexical item, and cannot look "into" sibling constituents. So, you can't write a subcategorization frame for a V that requires that it be followed by a PP containing another PP or something.

The Chomsky Hierarchy. Back at the beginning of computer science, Chomsky outlined a hierarchy of power between different ways that formal languages can be described. The PSRs we are using are a formal grammar of the sort he was talking about. Without getting into the details of the hierarchy, the point here is that any context-free grammar can be described as a context-sensitive grammar (with an unconstrained context), but not vice-versa. So, the context-sensitive grammar is more "powerful" (in a way that we don't really want). If we can describe language without the power of a context-sensitive grammar, this explains more. So, we are going to try to make do with a context-free grammar first, and back away from that only if we cannot do without context sensitivity. (And we will not be backing away from it in this class.)

1.4 Heads, projection of features

By adopting the constraints just mentioned, we put ourselves in a kind of a corner, because we know that there are verbs (like *give* vs. *put*) that "select for" different prepositions, but yet we didn't really have a way to distinguish in a subcategorization frame whether a PP was a *to*-PP or a *for*-PP (more generally, a locative PP).

(1) Marge put books on shelves

(2) * Marge put books to shelves

(3) * Marge gave books on Bart

(4) Marge gave books to Bart

We could do something like this:

⋮	
VP →	V NP PlocP
VP →	V NP PtoP
PlocP →	Ploc NP
PtoP →	Pto NP
on,	Ploc
to,	Pto
put,	V, [+ __ NP PlocP]
give,	V, [+ __ NP PtoP]
⋮	

But then we lose the connection *between* Ploc and Pto (they might as well be different categories altogether, like N and Adv). What we do instead is:

- Take the difference between *for* and *to* to be a property of each P.
- So, *for* has a [+loc] feature (as do other locative Ps).
- And, *to* has a [+to] feature (distinct from [+loc]).
- A PP headed by a [+to] P is itself [+to]. (The feature *projects* from P to PP.)
- A PP headed by a [+loc] P is itself [+loc].
- A subcategorization frame can refer to *features*.

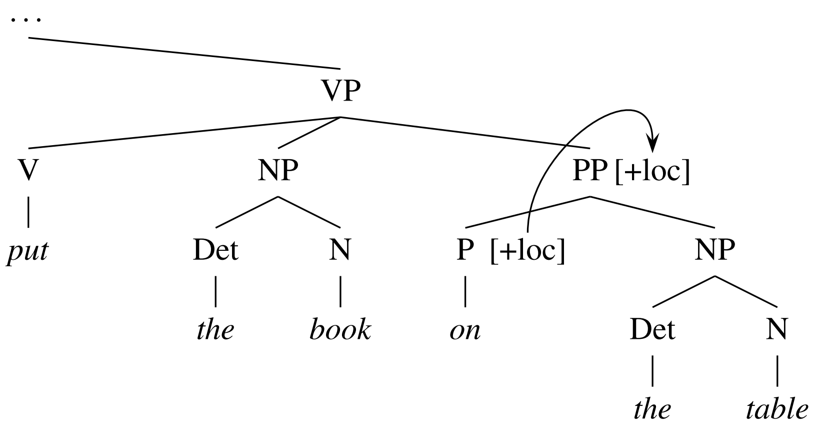
So what we end up with is:

⋮	
PP →	P NP
VP →	V NP PP
on,	P, [+loc]
to,	P, [+to]
put,	V, [+ __ NP $\text{PP}_{[+\text{loc}]}$]
give,	V, [+ __ NP $\text{PP}_{[+\text{to}]}$]
⋮	

Formalizing the "projection" of features, we say that:

Principle F: Features pass from a head to phrase it projects.

Specifically, when we have a "endocentric" (head-internal) rule like:
PP → P NP
The features of P move up to become the features of PP.



1.5 Subjects and objects

We can use feature projection to explain/predict the distribution of subject pronouns like *they* and object pronouns like *them*.

(5) They like them

(6) * They like they

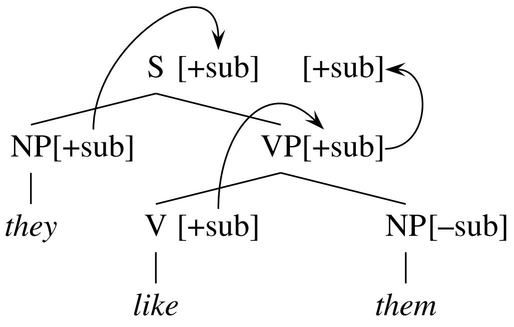
(7) * Them like them

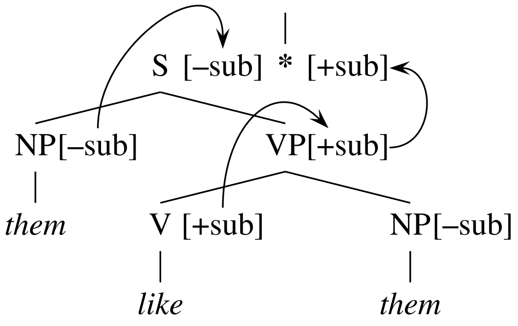
(8) * Them like they

⋮	
they,	NP, [+sub]
them,	NP, [–sub]
like,	V, [+sub], [+ __ $\text{NP}_{[-\text{sub}]}$]
⋮	

If transitive verbss have a subcategorization frame that requires a [–sub] NP object, then only *them* and not *they* will work as an object.

If we assume that since S → NP VP is *exocentric* (does not have an internal head), features are inherited *both* from NP and VP, and that it can't inherit [–sub] from the NP and a conflicting [+sub] from the VP and still be grammatical, then we can explain/predict why *them* is not allowed as a subject.





2 Subject agreement

We haven't talked about subject agreement yet, but we already kind of have it, once we've done the subject/object pronoun thing above.

(9) They like Bart

(10) * Lisa like Bart

(11) Lisa likes Bart

(12) * They likes Bart

We just need to assume that *Bart*/*Lisa* and *they* differ in number. The former are singular, the latter is plural. We can record this in the lexicon.

Lexical entries:

- *Bart*, NP,
- *Lisa*, NP,
- *they*, NP,
- *like*, V, [+ __],
- *likes*, V, [+ __],

3 Complements and adjuncts

Complements	Adjuncts
May be obligatory	Are always optional
Cannot be iterated	Can be iterated
Display lexical sensitivity	Are not lexically sensitive
Are sisters to the head	Are sisters to XP

3.1 Obligatoriness and iterativity

(13) Pat cut the bagel dramatically with a tiny knife

(14) Pat cut the bagel with a tiny knife dramatically

(15) Pat cut the bagel

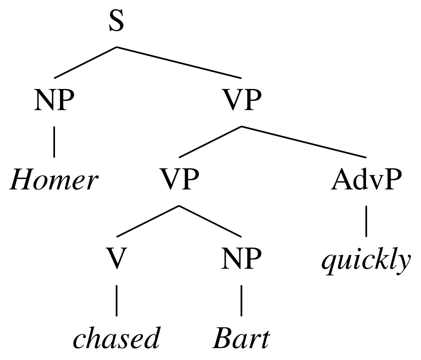
(16) * Pat cut

The things that are required are, generally, required *semantically* in order to "complete" the described event/state. The verb describes an event that relates a certain set of actors. Transitive verbs generally relate an agent of an action to a theme/patient of an action. Here is a table based on one in Larson (2010).

Role	Description
Agent	Volitional initiator of an action
Patient	Object or individual undergoing action
Theme	Object or individual moved/affected by action
Goal	Individual toward which action is directed
Source	Object or individual from which something is moved by the action, or from which the action originates
Experiencer	Individual (conscious) experiencing some event or state
Beneficiary	Object or individual that benefits from some action or event
Location	Place at which an individual, event, or state is situated
Instrument	Secondary cause of event; an object or individual causing some event through the action of an agent

There needs to be a match between the roles a verb needs and gets. The subcategorization frames in the lexicon encode basically this.

The things that are optional are the *adjuncts*. They are modifiers. They are not necessary but add information. Adverbs, adjectives, most PPs.



Can a complement come after an adjunct?

3.2 Lexical sensitivity

We saw an example of lexical sensitivity already—*put* requires a locative PP, *give* requires a *to*-PP. But *run*, for example, doesn't put any requirements on PPs. So, the former are complements, latter are adjuncts.

3.3 Examples

1. *John gave Ringo a drum on his birthday.*
   Complement(s):

   Adjunct(s):

   Arguments for the above divisions:

   Tree:

   PS Rules:

   Lexicon:

2. *Georgina walked to school nonchalantly*
Complement(s):

Adjunct(s):

Arguments for the above divisions:

Tree:

PS Rules:

Lexicon:

3. *River phrased her words in a strange manner*
Complement(s):

Adjunct(s):

Arguments for the above divisions:

Tree:

PS Rules:

Lexicon:

4. *Pat danced a jig near Chris*
Complement(s):

Adjunct(s):

Arguments for the above divisions:

Tree:

PS Rules:

Lexicon: