

CAS LX 522 Syntax I

6

Merge, feature checking
(3.6-4.2)

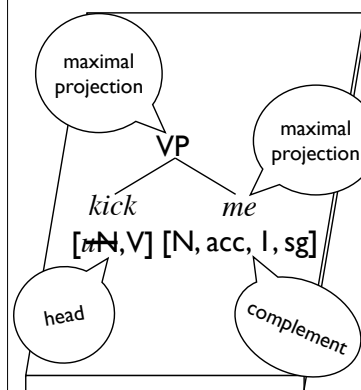
Syntactic operations

- **Merge** is a syntactic operation. It takes two syntactic objects and creates a new one out of them.
- The new syntactic object created by Merge inherits the features of one of the components (the head projects its features).
- Merge cannot “look inside” a syntactic object. Syntactic objects are only combined at the root.
- **The Extension Condition:** A syntactic derivation can only be continued by applying operations to the root projection of a tree.

Feature checking

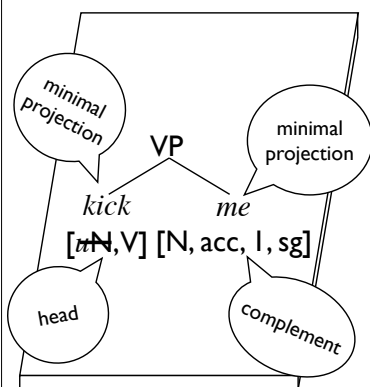
- Syntactic objects have features.
 - Lexical items (syntactic objects) are bundles of features.
- Some features are **interpretable**, others are **uninterpretable**.
- By the time the derivation is finished, there must be no uninterpretable features left (*Full Interpretation*).
- Uninterpretable features are eliminated by **checking** them against matching features. This happens as a result of Merge: Features of sisters can check against one another.
- Merge doesn't just happen. It *has* to happen.

Heads and complements



- When Merge combines two syntactic objects, one projects its features, one does not.
- When a lexical item projects its features to the combined syntactic object, it is generally called the **head**, and the thing it combined with is generally called the **complement**.
- A syntactic object that projects no further is called a **maximal projection**.
 - Where X is the category, this is alternatively called X^{max} or XP.
 - The complement is necessarily a maximal projection.

Heads and complements



- A syntactic object that has not projected at all (that is, a lexical item) is sometimes called a **minimal projection**.
 - Where X is the category, this is alternatively called X^{min} or X.
 - The head is a minimal projection.
 - In traditional terminology, the complement of a verb is generally called the **object** (or “**direct object**”).
 - So, often, is the complement of a preposition (“object of the preposition”).

Linear order

- Merge takes two syntactic objects and combines them into a new syntactic object.
- Merge does not specify *linear order* (which of the two combined objects comes first in pronunciation).
- In the English VP, heads always precede complements. But languages differ on this.

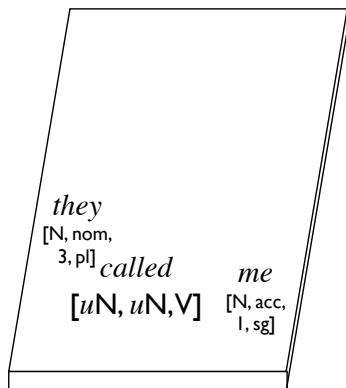
The head parameter

- Languages generally have something like a *basic word order*, an order in which words come in in “neutral” sentences.
- English: SVO
 - Akira ate an apple.
- Japanese: SOV
 - John wa ringo o tabeta.
John top apple acc ate
'John ate an apple.'
- In our terms, this amounts to a (generally language-wide choice) as to whether heads are pronounced before complements or vice-versa.
 - English: **head-initial** Japanese: **head-final**

Second Merge

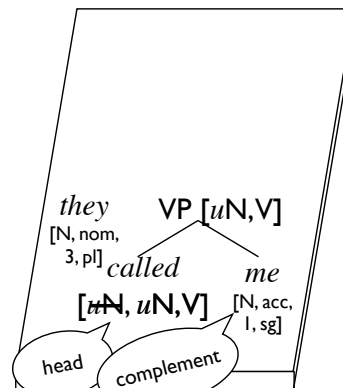
- Merge occurs when there is a selectional feature that needs to be satisfied.
 - If there is more than one such feature, Merge must happen more than once.
- As always, the node that projects is the one whose selectional feature was satisfied by the Merge.
 - The sister of the head (that projects) after the first Merge involving that head is called the **complement** (as above).
 - The nonprojecting sister of a syntactic object that has already projected once from a head is called the **specifier**.

Heads and complements



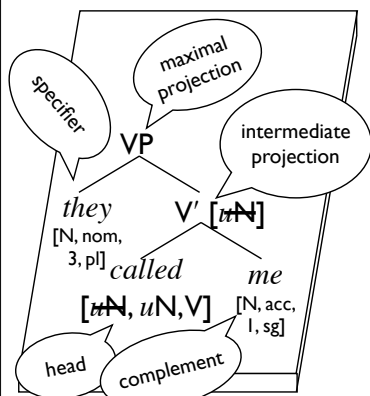
- A transitive verb like *called* needs two arguments (the caller and the callee).
- We encode this knowledge by hypothesizing two selectional features for N.
 - The first selectional feature will be checked by the callee.
 - The second selectional feature will be checked by the caller.
- So, *called* is Merged with *me*.

Heads and complements



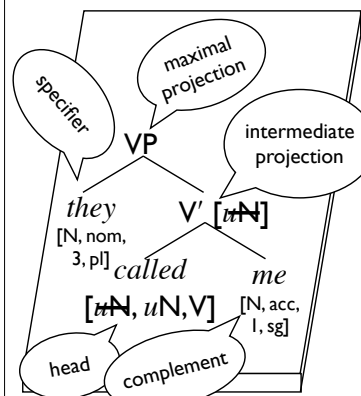
- So, *called* is Merged with *me*.
- One of the selectional features is checked off, the remaining features project to the new object.
- A selectional feature still remains.
- Merge applies again, Merging the new object with *they*.

Specifiers, XP, X-bar



- The second selectional feature has been eliminated.
- The sister to this second Merge is the **specifier**.
- A node that does not project further is a **maximal projection**.
- A node that has been projected and projects further is neither maximal nor minimal and is usually called an **intermediate projection**.

Specifiers, etc.

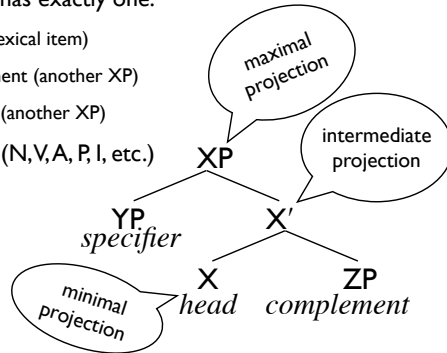


- In English, specifiers are on the left of the head, unlike complements.
- As with the head-complement order, languages (arguably) also differ in the linear order of their specifiers.
 - However, Spec-initial order is overwhelmingly more common...
 - VOS order (Malagasy)
Nahita ny mpianatra ny vehivavay.
saw the student the woman
'The woman saw the student.'

- In the '70s and '80s, these ideas went by the name "X'-theory":

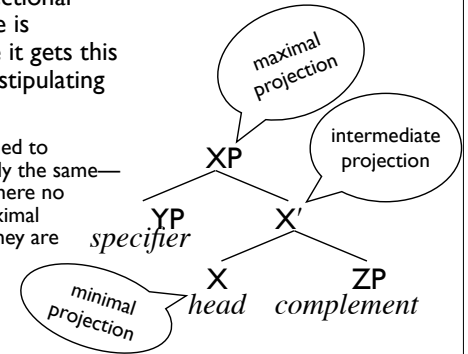
X'-theory

- Every XP has exactly one:
 - head (a lexical item)
 - complement (another XP)
 - specifier (another XP)
- for any X (N,V,A,P,I, etc.)



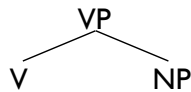
Merge v. X'-theory

- The system of selectional features and Merge is preferable because it gets this structure without stipulating the template.
- The structure assigned to sentences is generally the same—except that for us, there are no intermediate or maximal projections unless they are needed.



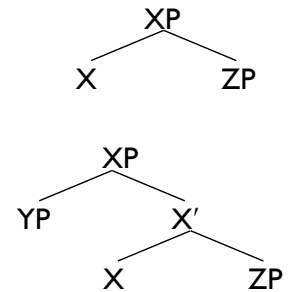
Node labeling conventions

- When we Merge two objects, the features of one of them project to become the features of the new object.
- The label for new node comes in two pieces:
 - The category (projected from the head)
 - The projection "level":
 - P = maximal projection
 - ° or nothing = minimal projection
 - ' = intermediate projection
- An XP is any node that does not project its features up.
- An X° (or X) node comes from the lexicon.

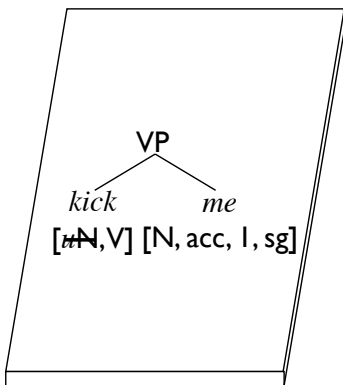


Maximal v. Minimal v. Intermediate

- Notice that whenever you Merge two things, the result is going to be a maximal projection. An "XP".
- But if in the next step it projects when you Merge it with something, that same node is now an intermediate projection.



Features and checking



- When we combine two things with Merge and check an uninterpretable feature, we cross it out.
- For simplicity, we can simply write the features under the head, and cross them out there.
 - This is as opposed to copying all but the checked feature and into a feature specification of the VP node.
 - This is just about how we write it down, it is the same system either way.

Adjuncts

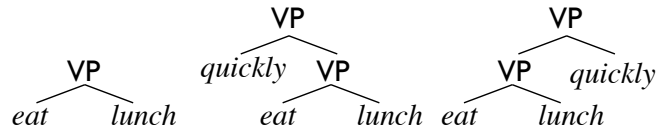
- *Pat put the book.
- Pat put the book on the shelf.
- Pat put the book on the shelf dramatically.
- Pat put the book on the shelf dramatically on Tuesday.
- Pat put the book on the shelf dramatically on Tuesday before several witnesses.
- Some things are required. Some things are not.
 - Arguments** get θ -roles and are **required**.
 - Adjuncts** are modificational and are **optional**.

Adjuncts and distribution

- Adjuncts are relatively “transparent”— having an adjunct does not seem to change the distributional characteristics.
 - Pat wants to eat lunch (quickly).
 - Pat wants to dine.
 - *I like to draw eat lunch (quickly).
 - I like to draw (happy) elephants.
 - *Pat wants to (happy) elephants.
- Idea: A verb (phrase) with an adjunct is still a verb (phrase), just as if it didn't have an adjunct.

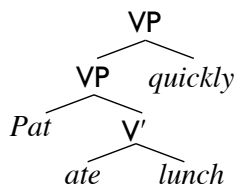
Adjoin

- The operations Merge and Adjoin are two different ways to combine two objects from the workbench.
- Merge takes two objects and creates a new object (with the label/features inherited from one of them).
- Adjoin attaches one object to the top of another one.
 - The linear order of adjuncts does not appear to be set parametrically, so they can either be before or after the object they attach to.



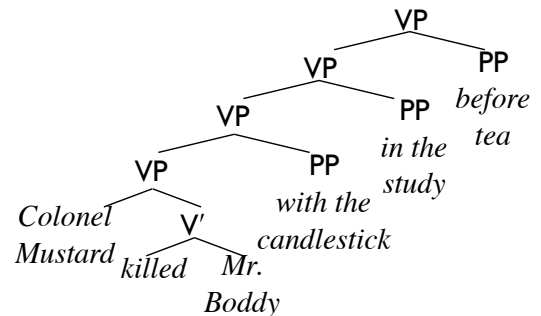
The luxury of adjunction

- We will also assume that Adjoin only applies to maximal projections.
- That is: If a syntactic object still has a selectional feature, Adjoin cannot attach something to it. Merge must happen first. Once all of the things that *need* to happen are taken care of, *then* you have the luxury of adjunction.



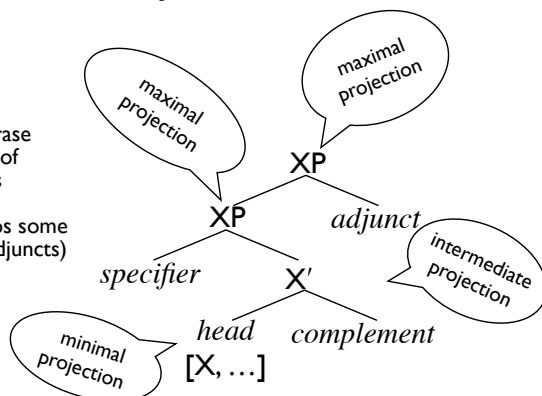
The luxury of adjunction

- Any number of adjuncts can be added, generally in any order. Adjuncts come in many different categories—“adjunct” is not a category, but rather a structural description.



A phrase

- So, a full phrase can have all of these pieces (plus perhaps some additional adjuncts)



Complements vs. adjuncts

- PPs seem to be freely reorderable— when adjuncts.
 - I ate lunch on Tuesday at Subway with Pat
 - I ate lunch on Tuesday with Pat at Subway
 - I ate lunch with Pat on Tuesday at Subway
 - I ate lunch on Tuesday with Pat at Subway
- But consider *glance at Chris*.
 - I glanced at Chris on Tuesday
 - *I glanced on Tuesday at Chris
- Ok: Why?

Mary saw him

- A pronoun like *him* refers to somebody in (our mental model of) the world.
- A pronoun refers to somebody or something that's been part of the conversation, or that you are pointing at.
- When you hear a pronoun and want to interpret it, you have to resolve its reference.

John arrived. Mary saw him.

- Here, *him* is likely to refer to John.
- Though we could be pointing at Bill, in which case *him* refers to Bill.
- The person who *hears* this has to figure it out.
- The person who *says* this knows who they meant.
- And had the grammar that generated the sentence.

Indices

- To describe what the speaker intended (that is, which sentence the speaker actually used), we use an *index* on each referent.
 - 1) John_i arrived. Mary_j saw him_i.
 - 2) John_i arrived. Mary_j saw him_k.
- The index represents what you are “pointing at” (perhaps just mentally).
- Two noun phrases that share an index *necessarily* share the same reference. They are coreferential.

Seeing him in the mirror

- Regard: Ike_i, Jim_j, Kristin_k.
 - 1) There's Ike_i. Kristin_k saw him_j in the mirror.
 - 2) There's Jim_j. Kristin_k saw him_j in the mirror.
 - 3) There's Ike_i. Jim_j saw him_i in the mirror.
 - 4) There's Jim_j. *Jim_j saw him_j in the mirror.
- What's wrong with that last one?

Seeing himself in the mirror

- Right, ok. Jim_j saw *himself*_j in the mirror.
- For some reason, when *Jim* is the subject and *him* is an object, *him* can't refer to *Jim*. Furthermore:
 - 1) Jim_j's father_k saw him_{i/j/*k} in the mirror.
 - 2) Jim_j's father_k saw himself_{k/*j/*i} in the mirror.
 - 3) Jim_j's father_k said that Mary_m saw him_{i/j/k} in the mirror.
 - 4) Mary_m introduced Jim_j to him_{i/*j}.
 - 5) Mary_m introduced Jim_j's father_k to him_{i/j/*k}.

Binding Theory

- **Binding Theory** consists of three Principles that govern the allowed distribution of NPs.
 - Pronouns: *he, her, it, she, ...*
 - Anaphors: *himself, herself, itself, ...*
 - R-expressions: *Pat, the student, ...*

R-expressions and anaphors

- 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.
- An anaphor does *not* get its meaning from something in the world—it depends on something else in the sentence.
 - 1) John saw himself in the mirror.
 - 2) 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 somewhere else.
 - 1) John told Mary that he likes pizza.
 - 2) Mary wondered if she agreed.
- ...but it doesn't *need* to be something in the sentence.
 - 1) Mary concluded that he was crazy.

Constraints on coreference

- 1) John_i saw himself_i.
 - 2) *Himself_i saw John_i.
 - 3) *John_i's mother saw himself_i.
- It is impossible to assign the same referent to *John* and *himself* in the (2) and (3). What is different between the good and bad sentences?

John's mother

- *John's mother* is an NP.
 - 1) [John's mother]_i saw herself_i.
 - 2) She saw John.
- But it's an NP that is made up of smaller pieces (*John's* and *mother*).
- So what is the internal structure of the NP *John's mother*?

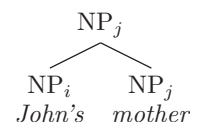
[_{NP} John's mother]

- Remember that pronouns come in three distinguishable forms (in English):
 - *I, he, she* nominative
 - *Me, him, her* accusative
 - *My, his, her* genitive
- The genitive case forms seem to have pretty much the same kind of "possessive" meaning that *John's* does.
- So, let's suppose that *John's* is the genitive case form of *John*.

[_{NP} John's mother]

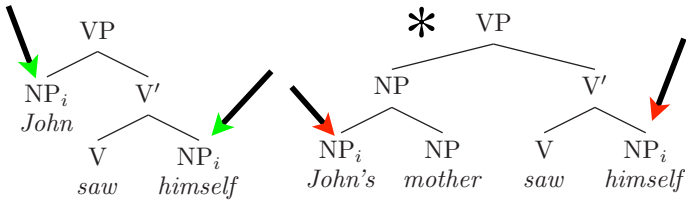
- Another point about *John's mother* is that it seems that the head should be *mother*.
- *John's* sort of modifies *mother*.
- Sort of like an adjective does... sort of like an adverb does for a verb...
- Let's suppose that *John's* is just adjoined to the NP *mother*.

- Only for now! To be revised in ch. 7.
- This is kind of hard to draw clearly.



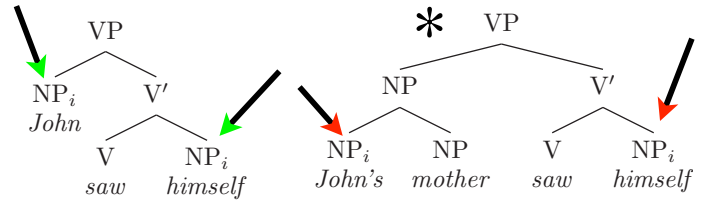
Command domains

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

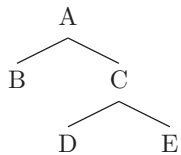


Command domains

- We think of the position that *John* occupies in in the first tree as being a position from which it “commands” the rest of the tree. It is hierarchically superior in a particular way. (Really, “non-inferior”)



Tree relations



- A node X c-commands its sisters and the nodes dominated by its sisters.
- B c-commands C, D, E.
- D c-commands E.
- E c-commands D.
- C c-commands B.
- A c-commands nothing.