Engl.336 Syntax

Additional notes for Chapter 3

We ended our discussion with our definition of Merge, the basic syntactic operation that builds up constituents by joining two objects together. Here are the properties of Merge that we've discussed so far:

— Merge is binary; that is, the operation can only join two objects (or constituents).

— One of the two objects will be the head; that is, it will be the object whose features project upwards to the resulting object of Merge (we will further discuss headedness below).

— Let's say that we have this tree diagram of a Merge operation joining Y and Z:

— Each line between ZP, the syntactic object created by Merge, and the syntactic objects Y and Z, is called a branch. Branches denote that one object contains or dominates the other. So, in this instance, ZP contains or dominates both Y and Z.

— The objects connected by the branches are called nodes. The topmost node of the structure is called the root node (think of it as an upside-down tree (diagram)), while the lowest nodes are called terminal nodes.

— Remember that Merge doesn't care about the order of the syntactic objects that it joins. In other words, the Merge of Z and P or P and Z will derive the same syntactic object.

— Remember also that syntactic objects can only be merged at the root nodes. So, if we were to Merge the structure above (the tree diagram) with another object, it would be Merged at the root node, ZP, and not Y or Z.

— Heads and headedness:

If we observe enough syntactic constituents in natural language, we can notice a pattern where one element constrains the distribution of the constituent, sets the agreement relation with other constituents and serves as the locus of inflection, and determines the reference of the constituent. Let's take these one by one:

1- Constrains the distribution of the constituent: Let's say that we have the constituent "cats" (remember that the smallest constituents are lexical items). Let's see how it patterns in different syntactic constructions:

A — I saw cats.

B — Cats hunt mice.

C — Cat are predators.

D — *I prayed cats.

What we're seeing here is the fact that "cats" can occur in certain positions and in certain syntactic structures only, and this applies to every other lexical item (we'll discuss more of this in the coming lecture(s)). We can say, then, that the distribution of "cats" is somewhat limited.

Let's try with "beautiful cats," another constituent.

A — I saw the beautiful cats.

B — Beautiful cats hunt mice.

C — Beautiful cats are predators.

D — *I prayed beautiful cats.

Note how, even despite having a different structure, the constituents "cats" and "beautiful cats" follow the same distribution. You can add a million other adjectives before "cats" and the result would still be the same. You can also try with "these cats," "some cats," and even "some of these beautiful cats"; the result would still be the same. This tells us that there is something about "cats" that determines the distribution of the above constituents, and that property is headedness.

2 — Sets the agreement relations with other constituents and serves as the locus of inflection:

Let's, again, take "beautiful cats" to be a constituent in, say, "I saw beautiful cats" (remember that constituents can be identified by replacement/substitution, so beautiful cats are a constituent in this sentence because we can replace them with "them"). Observe the following structures:

A — The beautiful cats are sitting on the fence.

B — The beautiful cat is sitting on the fence.

Note how, with the change of the constituent, the auxiliary verb changes or inflects to match its the features. This is also observed even if we change the adjective modifying "cats," as follows:

C — The creepy cats are sitting on the fence.

D — The creepy cat is sitting on the fence.

Again, note how the auxiliary verb inflects to match the features of the constituent. Despite changing the adjective, it is still “cat(s)” that sets or determines agreement relations between the constituent and other elements in the sentence. In this instance, “cat(s)” is the head in the above constituents.

A closely related property of heads is that they serve as the locus of inflection. In other words, heads are the items bearing inflections marking syntactic relations with other constituents. Observe the following:

A — There is a creepy cat sitting on the fence.

B — There are two creepy cats sitting on the fence.

Note how it is the head of the constituent, “cat(s),” that bears the inflection marking for plurals, not the adjective, “creepy.”

3 — Determines the reference of the constituent:

The reference of the constituent “creepy cats,” that is, the real-life object or entity that is picked out by the constituent or the one which the constituent stands for, is mainly determined by the head, “cats.” Think of it this way: even if we string along a thousand adjectives before “cats” in this constituent, we will still be referring to a cat. However, if we change the head, replacing it with pigs, for example, the entity that the constituent refers to will be fundamentally different. So, it is indeed heads that (mainly) determine the reference of the constituent.

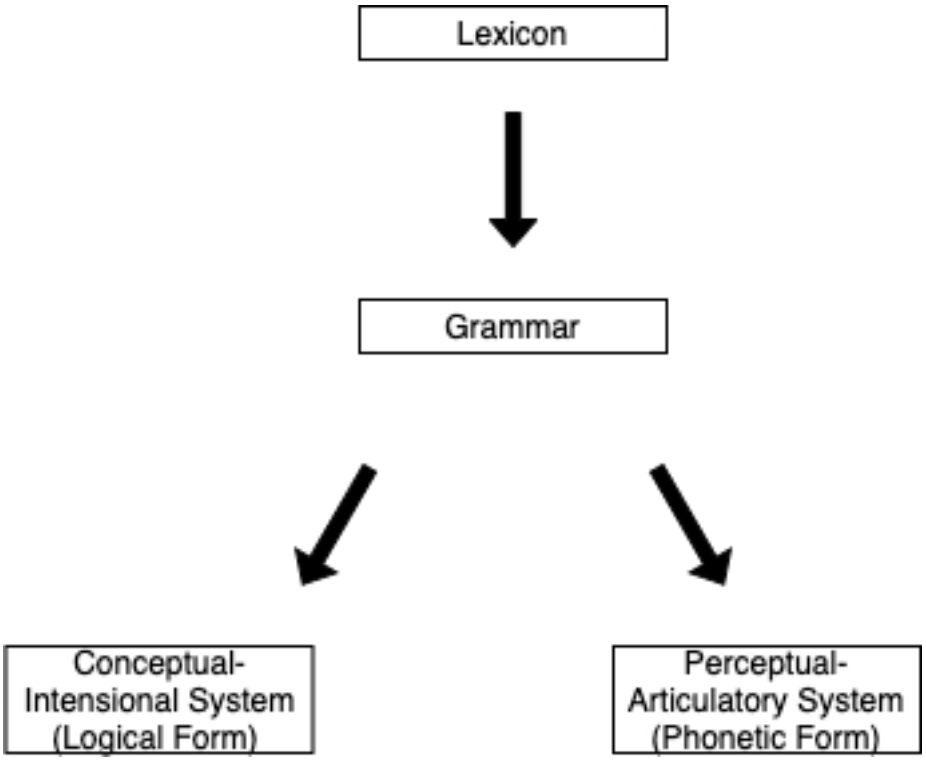
So, in summary, heads have these three particular properties: they determine the distribution of the constituent in the sentence, they determine the agreement relations between the constituent and other elements in the sentence and serve as the locus of inflection, and they determine the reference of the constituent.

But what do heads have to do with Merge, you might ask. Well, everything!

— Syntax, the LF, the PF, derivations, and (un)interpretability

In our discussion on the autonomy of syntax, we described the creation of well-formed sentences by the syntax as its own autonomous process that draws items from the lexicon, which is the list of lexical items stored in the minds of speakers or users of a language, applies syntactic operations on them, such as Merge. The result of this process, which is termed a derivation, are syntactic objects that then interface with the conceptual-intentional system, which is concerned with meaning and understanding in the mind, through the Logical Form (LF), and the articulatory-perceptual system, which is concerned with speech and gestures, through Spell-out/Phonetic Form (PF)[1].

[1] A set of operations apply to the syntactic object, and the representation that arises out of these operations is the Phonetic Form that interfaces with the articulatory-perceptual system. So, in our view, Spell-out is a point in which a set of operations apply to a syntactic object.



At LF, the syntactic object must "consist entirely of of legitimate objects" (Chomsky, 1995, p. 178); that is, the object must not have any uninterpretable features as presented in the Full Interpretation constraint in our textbook (p. 66), reproduced here:
Full Interpretation: the structure to which the semantic interface rules apply contains no uninterpretable features.

Uninterpretable features, as we discussed before, are thought of as devoid of semantic content, while interpretable features carry semantic content. Examples of interpretable features are phi-features, [Gender], [Number], and [Person]. Examples of uninterpretable features include [Case]. Think of it this way: interpretable features are those that can be interpreted at LF, which is the interface level to the conceptual-intenstional system, meaning that these features affect our own semantic interpretation of the lexical item to which they belong. Uninterpretable features, like Case, do not affect this interpretation. Whether the Case feature specified on the lexical item is accusative [ACC], nominative [NOM], or genitive [GEN], this wouldn't really affect our semantic interpretation of this specific lexical item; boiled down to its bare essentials, Case is (roughly) the grammatical marking of the syntactic position of an item.

So, to summarize, items are specified with interpretable and uninterpretable features, and the syntax operates on them with syntactic operations such as Merge. In order for the resulting syntactic object to be interpretable at LF, the interface level with the conceptual-intensional system, any uninterpretable features specified on the syntactic object have to be deleted, and this is done through Merge, whereby matching hereby matching features are checked and then deleted. In other words, a head, with an uninterpretable feature, Merges with another object that has a matching interpretable feature. These features are then checked and deleted, and the resulting syntactic object is thus free of any uninterpretable features; the Full Interpretation constraint has been met.

So this is how heads relate to Merge: heads are specified with an uninterpretable feature that has to be checked by Merging with another object that bears a matching interpretable feature. Matching here means of the same type or kind. So, for example, an object specified with an uninterpretable verb feature [*u*V] has to Merge with another object that bears a matching interpretable verb feature [V] in order to check and delete the uninterpretable feature.

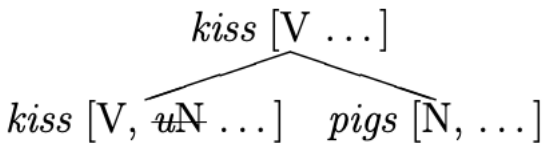


Figure 1

In Figure 1, above, "kiss" has an interpretable category feature, [V], and an uninterpretable noun feature, [*u*N], meaning that it needs to Merge with another object that has a matching interpretable feature, and this object is "pigs," which has a matching interpretable [N] feature. The uninterpretable feature is checked and marked for deletion. Note also that it is the head that has the uninterpretable feature, meaning that it is the head that triggers Merge to check its uninterpretable feature(s). In other words, it is the head that selects a syntactic object that has matching interpretable features. The uninterpretable features on heads are also called c-selectional features, or category selectional features, and it is to these that we now turn. But first, let's discuss theta-roles.

— Propositions, predicates, and theta-roles:

In order to motivate the argument that lexical items need to combine or Merge with others, we will have to discuss some ideas and notions in semantics. First, we have the idea of concept. Each lexical item expresses a certain concept, be that property, like **buy** and **sell**, or movement, like **run** and **walk**. Some of these concepts are ever-present in communication, so their meaning was fossilized in the respective lexical items. This fossilization is called lexicalization, and the lexicalized words are called predicates.

Predicates, like we discussed before, attribute a certain quality to entities or denotes the relationship between two or more arguments. The combination of predicate and argument is known as a proposition. Put more technically, a proposition comprises an argument (or more) and a predicate. So, these predicates that are lexicalized into a certain language need arguments — entities — in order to form a proposition.

Predicates that need one argument are called 1-place predicates; predicates that need two arguments are called 2-place predicates; and predicates that need three arguments are called 3-place predicates. There doesn't seem to be a language which lexicalizes predicates that require more than three arguments.

There is a type of predicate that doesn't really need an argument: 0-place predicates, like verbs describing the weather, such as rain, snow. For example:

• It rained
• It snowed

You might think that there is an argument in the above proposition: "It." The pronoun here doesn't really function as an argument; it doesn't really add anything semantically important to the proposition. Note that a sentence like

• *The weather rained

is ill-formed, or at least not accepted by the majority. That is because the predicate "rain" is a zero-place predicate and "the weather" is an argument that adds semantic value to the proposition.

— Theta-roles:

Predicates seem to classify the arguments they take into certain semantic types. For instance, one-place predicates like "ran," "joke," and "galloped" seem to combine with expressions that initiate the action described by the verb:

• Allison ran
• Allison Joked
• The horse galloped

These expressions play the role of the Agent (or Actor or Causer).

Now, look at these examples and note the difference:

• Allison fell
• Allison collapsed
• The horse appeared

The expressions that combine with these predicates play the role of Theme. Predicates which combine with an Agent are called unergatives and those that combine with a Theme are called unaccusatives. Predicates that combine with both an Actor and a Theme are called transitives.

These properties of predicates, that they combine with expressions playing certain roles or having certain semantic features, are called thematic roles. These roles are part of the lexical semantics of predicates. Each predicate needs a certain number of expressions with certain semantic content. In other words, each predicate has a certain number of thematic roles, or theta roles (θ-roles) to assign. One-place predicates assign one θ-role and are called intransitives; two-place predicates assign two θ-roles and are called transitives; and three-place predicates assign three θ-roles and are called ditransitives.

(1) Ahmad gave the car to Mohammad.

(2) Mohammad gave the money to Ahmad.

In (1), Ahmad is the Agent, the car is the Theme, and Mohammad is the Goal (the expression towards which the action denotes by the verb unfolds). In (2), Mohammad is the Agent, the money is the Theme, and Ahmad is the Goal.

While θ-roles are semantic in nature, they do have an effect on syntactic structures.
For instance, if a predicate has one θ-role that isn't assigned to an expression, the syntactic structure is deemed ill-formed:

• *Ahmad gave

• *Jonathan addressed

This data suggests that each predicate has to assign all the θ-roles that it can assign. Moreover, each constituent can only be assigned on θ-roles. This leads us to make a generalization.

The Unique θ Generalization: Each θ-role is assigned to exactly one constituent in the sentence.

Note, however, that this doesn't mean that all constituents have to be assigned a θ-role, but that the predicate has to assign all the θ-roles it can assign, each to only one constituent. These expressions that are assigned θ-roles are called arguments.

Note also that there are some contexts in which predicates don't assign all of their θ-roles, at least not overtly:

• He donated $150,000.

Here, the verb donate is a three-place predicate, assigning an Agent, Theme, and Goal θ-roles. In the above sentence, however, we only see two θ-roles, the Agent and the Theme, and yet this is a well-formed sentence, albeit in specific contexts. The well-formedness of this sentence depends on the context in which it was uttered. If the Theme been previously mentioned, then it would be perfectly okay for the predicate to assign two overt θ-roles, with the third being assigned to the topic of the discussion, but with no phonological expression.

Other verbs alternate between being one- and two-place predicates based on the intended meaning. Eat, for example, can be a one-place predicate in the following example:

• He ate this morning.

And a two-place predicate:

• He ate a sandwich.

Again, depending on the meaning that is intended by the speaker.

— Theta-roles and syntactic features:

There is a strong relationship between θ-role and syntactic requirements, as we saw above, but that doesn't mean that we can predict the syntactic category from the assigned θ-role. For example:

• I felt ill
• I felt a slight bump.
• I felt that I was about to pass out.

Here, the pronoun is assigned the θ-role of Experiencer, while "ill," "a slight bump," and "that I was about to pass out" are all Theme or Source. However, these constituents have different syntactic categories: adjective, NP, and sentence, respectively.

The argument goes the other way too, meaning that we can't predict the assigned θ-role from the syntactic category:

• Allison ran.
• Allison collapsed.

Here, the same noun, Allison, is Agent in the first sentence and Theme in the second.

But what is, then, the relationship between θ-roles and syntactic category? How does our theory of syntax account for the fact that some verbs take certain distinct arguments rather than others? Take the following data, for example:

• Allison kicked the stone.
• ?The stone kicked Allison.
• ?Allison kicked the meeting.
• Allison saw that the boy ate chocolate.
• *Allison kicked that the boy ate chocolate.
• He snored.
• He snored all night.
• *He snored the meeting.

What we're seeing here is that the predicates not only assign a certain θ-role, but also require a the arguments to be of a certain syntactic category. The objective is, then, to understand the system through which these θ-roles and syntactic categories are assigned (or selected) by the predicate.

— Category and semantic selectional features:

We can describe how certain lexical items Merge with other semantically and syntactically distinct items through features: categorical selectional features (c-selectional) and semantic selectional features (s-selectional features).

We first begin with c-selection features, also referred to as subcategorization (sub-cat) features.

C-selection features are uninterpretable features on lexical items that describe not its distribution, but the syntactic category of the lexical items with which it is able to Merge. So, in addition to the main category feature of a certain lexical item, it also has the uninterpretable c-selectional feature that determines what lexical items it can Merge with.

For example, the sentence “Touch the sky” is perfectly well-formed, but “Touch eat” isn’t, because the uninterpretable c-selectional feature on the predicate “Touch” is [*u*N], meaning that it can only Merge with a noun, while the category feature of “eat” is [V].

So, we’ve established that lexical items have, among others, categorical features and uninterpretable sub-cat features. In order for the completed derivation — the structure after all syntactic operations are applied — to be passed onto the LF for semantic interpretation, it has to be free of uninterpretable features. This is captured in the Full Interpretation constraint that we’ve discussed above. The fact that uninterpretable features are checked and then deleted under Merge is captured by the following:

The Checking Requirement: Uninterpretable (c-selectional) features must be checked, and once checked, they can delete.

Checking under Sisterhood: An uninterpretable c-selectional feature F on a syntactic object Y is checked when Y is sister to another syntactic object Z which bears a matching feature F.

— S-selectional features:

S-selectional features define the semantic features of lexical items with which the head can combine. S-selectional features include: entity, property, and proposition.

- Proposition: an argument plus a predicate; in other words, the expression of a certain attribute or quality to an argument. Remember that propositions are expressed by sentences and can be verified, i.e. can be true or false. An example of a sentence expressing a proposition is “She is happy.”

- Entity: here, entity refers to living or non-living objects.

- Property: an attribute, such as “happy” or “green.”

Different lexical items have different s-selectional features that constrict their combinations. For instance, the verb eat has an s-selectional feature only of entity, while say can have an s-selectional feature of proposition. Observe the following examples:

• Eat a cake.
• *Eat that she is happy.
• Say a prayer.
• Say that she is happy.

Despite their importance for determining the well-formedness of sentences, s-selectional features don’t figure in the syntactic combination process. In other words, Merge doesn’t inspect s-selectional features.

You might be wondering how all of this relates to Θ-roles. It turns out that c-selectional features and s-selectional features are associated with certain Θ-roles in the entry for each lexical item in the lexicon. Indeed, we can build a Θ-grid for each lexical item that includes the Θ-roles that it assigns and the associated c- and s-selectional features.

Meaning that, when a predicate assigns Θ-roles, it is assigning them to constituents that have the categorical and semantic features that match its uninterpretable features. We thus have a theory of why certain lexical items combine with others that have particular syntactic and semantic features. In other words, we have provided a syntactic implementation of the Unique Θ Generalization. Note that it is the head that has the uninterpretable features and it is the head that selects, but the primary property of heads is that hey project their features.

Note that Merge and Checking only occur on root nodes. Meaning that a derivation can only be continued with operations occurring on the root node. This is captured by the extension condition:

The Extension Condition: A syntactic derivation can only be continued by applying operations to the root of the tree.

— First and second Merge; phrase structure:

The structures generated by the application of the syntactic operation Merge are organized into phrases, derived from the selectional properties of heads. In other words, the head in a Merge operation projects its features upwards through the generated structure.

Structures which don't project features — those that had their c-selectional features checked — are called maximal projections, XP. In other words, a phrase of category X. For example, let's say that a lexical item with a categorical verb feature and a sub-cat noun feature, [V, *u*N], Merges with a lexical item with a categorical noun feature, [N]. Since the Merge operation would lead to the uninterpretable feature being checked and deleted, the generated structure has no remaining uninterpretable features, meaning that it is a maximal projection, with the label VP that is derived from the head.

A maximal object that is nominal, that is generated by the Merge of a head with a categorial noun feature, is NP, and a maximal object that is prepositional is PP, for instance.

[Figure 2]

Note that lexical items with no features to be checked are both maximal and minimal phrases. Maximal because there are no c-selectional features, and minimal because they are lexical items. In essence, the phrasal status of nodes is determined by their c-selectional features. Also note that c-selecting nodes, or heads, are sisters to maximal projections, since the items with which heads Merge can't have c-selectional features of their own. Otherwise, the uninterpretable feature of the head's sister node wouldn't be checked, leading to a structure with an uninterpretable feature at LF. Note that the sister of a head is called a complement, or an object in traditional grammars of English.

The structure where a complex phrasal object Merges with a lexical head is known as a head-complement structure, which arises from the first application of Merge.

In English and Arabic, for example, the complement Merges to the right of the head that selects them, but Japanese, for instance, has its complements to the left of their heads. For example:

Hanako ga Taro o tataku
Hanako subj Taro obj hit
'Hanako is hitting Taro.'

Languages like English are called VO (verb-object) languages, while those like Japanese are called OV (object-verb) languages.

— Second Merge and specifiers.

What happens, however, if the verb has two theta-roles to assign? In other words, the verb has two c-selectional features. Well, these features have to be checked so that the sentence can be derived.

Let's say that we have the sentence "Paul burns letters to Peter." "burn" actually has two theta-roles to assign. The first is the Agent theta-role and the associated c-selectional feature that needs to be checked, and the second is the Theme theta-role with its associated theta-role that needs to be checked.

After "burn" merges with the NP "letters to Peter," one of its c-selectional features are checked, but what happens with the other c-selectional feature? Well, it is projected upward through the generated structure, which is called an intermediate projection. An intermediate projection is created if the Merge operation hasn't checked all the uninterpretable features on the head. It is called $\bar{X}$ or X-bar, where X stands for the label projected by the head.

This intermediate projection then Merges with a maximal projection that has an interpretable feature that can be checked against the uninterpretable feature projected from the head, thus checking all of the uninterpretable features in the structure and generating a maximal projection, an XP. Much like the object that heads Merge with in the first application are called complements, the objects that $\bar{X}$ Merges with are called specifiers.

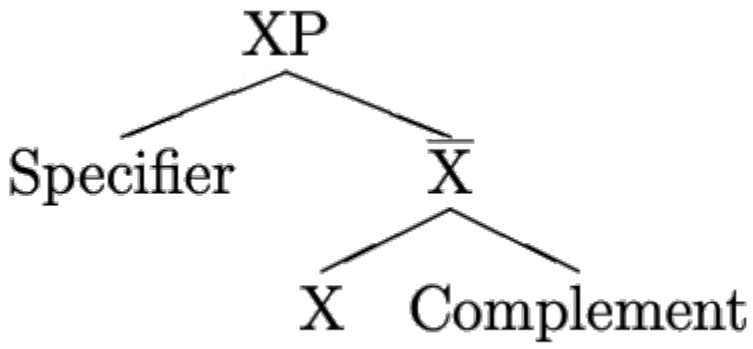


Figure 3