Constraining Merge: Selection and uninterpretable features

Introduction to Syntax, EGG 2011, Lecture 4

July 28th, 2011

1 θ-roles and selection

Let's think a bit about semantics.

- We've said that complete sentences or clauses generally express **propositions**.
- **?** But what about smaller pieces like phrases and words?
- At least some phrases seem to refer to entities in the world, like *that guy over there* or *those pigs*.
- So if we take such a phrase out of a sentence, what's left behind should mean something like a proposition that's missing a piece.

A proposition that's missing something like this is a **predicate**.

- Predicate is a very general term, used (with more or less the same meaning) in logic, mathematics and computer science in addition to linguistics.
- In language, predicates usually show up as verbs, verb phrases, adjectives or (certain kinds of) nouns.
- E.g. *love truffles* is a predicate, because if we put it together with *that guy over there* or *those pigs*, we get a proposition that could be either true or false.

Now, different verbs (and other predicates) have different numbers of missing pieces or **arguments**:

(1) a. Alejandro arrived.
    b. * Alejandro arrived the pigs.
(2) a. * Alejandro delivered.
    b. Alejandro delivered the pigs.
(3) a. * Alejandro put.
    b. * Alejandro put the pigs.
    c. Alejandro put the pigs in the sty.

- ➪ *arrive* is one-place or **intransitive**.
- ➪ *deliver* is two-place or **transitive**.
- ➪ *put* is three-place or **ditransitive**.

Predicates also place requirements on the kinds of arguments they take:

(4) Alejandro loves truffles.
(5) # The paint loves truffles.
(6) # Rosa sprayed Alejandro on the wall.
(7) Rosa sprayed the paint on the wall.

- The subject of *love* must be a sentient being, capable of experiencing emotion.
- The object of *spray* must be a liquid.

We say that each predicate assigns a certain number of **thematic roles**, one for each argument.

- We can try to categorize the restrictions on the arguments by identifying distinct roles, like AGENT, THEME, EXPERIENCER...
- Such role types are essentially semantic, but at least some of the information is syntactic, including the number of roles assigned and the categories of the arguments.
- When we refer to the syntactically relevant properties of thematic roles, we usually use the shorter term θ-roles.

Here's a plausible hypothesis:

(8) **The Unique Θ Generalization** Each θ-role of each predicate in a sentence must be assigned to an appropriate argument, but a constituent cannot be assigned more than one θ-role.

This additionally rules out things like 9 under the interpretation where *Alfons* accuses himself:

(9) * Alfons accused.

But 8 doesn't say that all constituents have to get a θ-role, because many clearly don't:

(10) Alejandro arrived **on his scooter**.

Predicates have more to say about their arguments than just their thematic roles:

(11) Murugan felt a feverish sensation.
(12) Murugan felt feverish.
(13) Murugan felt he had a fever.

☞ *feel* takes an EXPERIENCER subject and a THEME object.

☞ But the object can be a noun phrase, an adjective or an entire clause.

Other predicates are more particular:

(14) Murugan perceived a feverish sensation.
(15) * Murugan perceived feverish.
(16) Murugan perceived he had a fever.

(17) # Murugan became a feverish sensation.
(18) Murugan became feverish.
(19) * Murugan became he had a fever.

(20) * Murugan thought a feverish sensation.
(21) * Murugan thought feverish.
(22) Murugan thought he had a fever.

The term we use for this phenomenon is **c-selection** (short for category selection) or **subcategorization**.

- C-selection seems to be a purely syntactic matter, since it cannot be derived completely from the semantic properties of predicates and their arguments.
- So when we describe the argument-taking properties of a given predicate in our theory, we'll need to include several different kinds of information.

2 Uninterpretable features

How do we encode c-selection in our grammar?

- We said last time that we want to represent everything in terms of features, and c-selection should be no different.
- But the c-selectional properties of a given lexical item don't tell us about a property directly observable on the item itself.
- Rather, they say something about how it fits into a sentence, what sorts of things it can or must combine with.

➪ We need a special kind of feature for this.

Interpretable features like number on a noun or tense on a verb give us information about properties of the word itself, most typically ones that have a clear meaning.

Uninterpretable features give information about the contexts in which syntactic objects can occur. They aren't associated with any particular meaning, but act as instructions for putting sentences together in the right way.

We will indicate the difference by marking uninterpretable features with a *u* in front of them:

(23) X [G, *u*F]

So G is interpretable, F is uninterpretable.

Now we need to set things up so that uninterpretable features can do some work. Here's the first step:

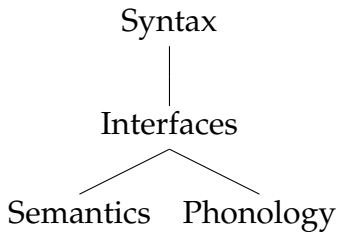
(24) **The Principle of Full Interpretation** The structure output by the syntax may not contain any uninterpretable features.

- This is nothing more than a formal statement about features that are meant to trigger syntactic operations.
- The real motivation is simply the assumption that we want to use features to implement syntactic requirements.
- A sort of conceptual underpinning for all of this is often assumed, but it is not strictly necessary or directly motivated by anything we've seen so far.

The idea is this:

- The output of the syntax is the input to the semantics
- Uninterpretable features are literally uninterpret**able**, not just uninterpreted so they are not allowed to be around when the semantics goes to work.
- Thus the derivation actually needs to eliminate them somehow beforehand.

This is further based on assuming something like the following as the shape of the derivation:



- The syntax creates structures and then sends them off to the two interfaces.
- The semantic interface determines a meaning for the structure, while the phonological interface determines a pronuncation.

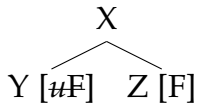
Whatever the motivation, we need a way to get rid of uninterpretable features in the syntax, before they get to the interfaces.

(25) **The Checking Requirement** Uninterpretable features must be **checked**, and once checked, they can delete.

(26) **Checking under Sisterhood** An uninterpretable feature F on a syntactic object Y is checked when Y is sister to another syntactic object Z which bears a **matching** feature F.

- E.g. Y in 27 has an uninterpretable feature [*u*F], so if left unchecked it would lead to ungrammaticality.
- We can merge Y with Z, which has a matching interpretable feature F.
- Now Y and Z are sisters, so [*u*F] can check, and everything turns out ok.

(27) Y [*u*F]

(28)


What's the point of all this?

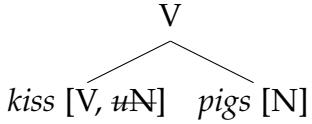
☞ We can now use uninterpretable features as instructions, triggers for appropriate instantiations of Merge.

☞ If a syntactic object doesn't Merge with the sort of thing demanded by its uninterpretable features, the derivation will crash, i.e. it will fail to derive a grammatical sentence.

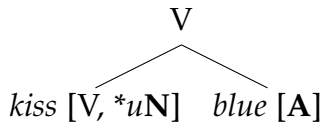
☞ This ensures that only those derivations succeed in which the right sort of things have Merged.

Specifically, we can now encode c-selection with uninterpretable category features:

(29) *kiss* [V, *u*N]

(30)


(31)



- So our theory correctly rules out sentences where the wrong category of argument combines with a predicate.
- It also rules out sentences where a predicate doesn't combine with enough arguments.
- Either way, an unchecked uninterpretable category feature will be left over at the end, causing a crash.

In addition to c-selection, we also need s-selection.

- This is where we encode the requirements a predicate places on the **semantic** type of its arguments.
- E.g. the object of *ask* can be of various syntactic categories, but it has to be semantically a question or piece of information that can be queried.

We won't really worry about s-selection, but you should know that it exists and seems to be independent of c-selection.

3 Back to heads

We can bring this all together to explain the determination of the head in a given phrase:

(32) **Definition of Head** The head of a phrase is the syntactic object which selects the other object which it Merges with to create the phrase.

➪ So the object that has an uninterpretable category feature checked off in the Merge process is the head.

And we can set down the importance of being the head:

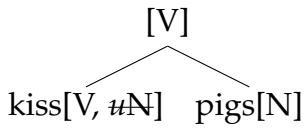
(33) **Headedness** The item that selects is the item that projects.

- Imagine that object X selects object Y, merging with it to create object Z.
- The further properties of object Z will be projected from the head, object X.

An example:

- The constituent *kiss pigs* is headed by *kiss*, because *kiss* selects a noun like *pigs*.

(34)



- So *kiss pigs* is essentially verbal, as *kiss* is verbal, and has a distribution related to verbs, not nouns:

(35) a. I want to [$_V$ sing].
b. I want to [kiss pigs].

(36) a. I want [$_N$ pigs].
b. * I want [kiss pigs].

The way things are set up lets us derive an interesting corollary:

(37) **Ban on Unchecked Features on Non-heads** If X selects Y and the two Merge, Y cannot have any unchecked uninterpretable features.

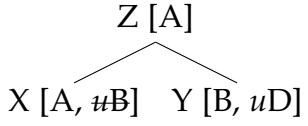
- In other words, only the head can have unchecked features.

Consider why this is:

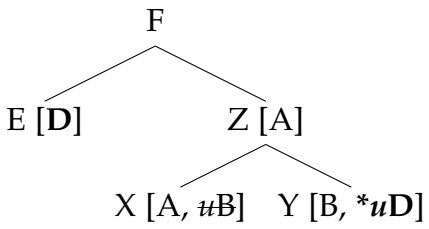
- When X and Y Merge, the features from X will project to the newly created constituent, but the features of Y won't.
- When this merges with something else, the features projected from X can be checked, but those on Y can't, because Y won't be the sister of the newly merged object.
- Any uninterpretable features on Y will thus remain forever unchecked, leading to a crash.

In structural terms:

(38)



(39)

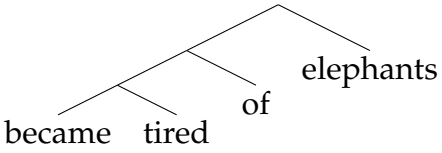


There is evidence that this is actually correct. Consider:

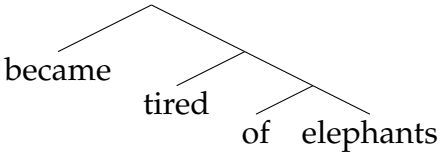
(40) Sandhya became tired of elephants.

- The verb *become* c-selects for an adjective, and the adjective *tired* c-selects in turn for a preposition, and the preposition *of* c-selects for a noun.
- 40 has all the right things for those requirements to be satisfied, but we could imagine them being combined lots of different ways.

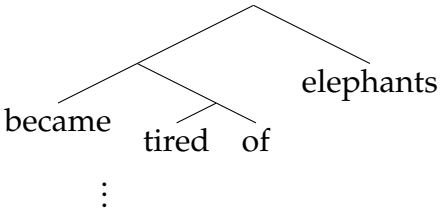
(41)



(42)

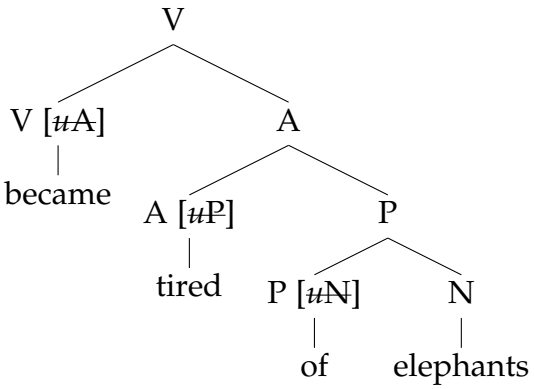


(43)



But the ban in 37 predicts that only one structure is possible: the one where the selectional feature on each object is checked before it itself is selected:

(44)



This is a good result, because constituency tests pick out the same structure. E.g.:

(45) [Tired of elephants] is something Sandhya will never become.
(46) * [Become tired] is something Sandhya never will of elephants.