Lecture 21: Agreement.

Andrei Antonenko

LIN 311: Syntax

November 6, 2018
Outline

1 Agreement
   Review of the Modules So Far
   Subject-Verb Agreement
   Features and Interpretability

2 Formal Theory of Features
   Types of Features
   Feature Checking
Agreement
There are two modules which we so far considered which limit X-bar theory:

1. **$\theta$-theory**
   - Interface with semantics.
   - Verb (and possibly noun) meaning determines how many arguments are there.
   - Rules out things like:

   (1) a. *Sally kissed.
   b. *Sue danced waltz tango.

2. **Case Theory**
There are two modules which we so far considered which limit X-bar theory:

1. **θ-theory**
2. **Case Theory**
   - Purely syntactic module.
   - Responsible for the correct forms and positions of DPs.
   - Rules out things like:

   (2) a. *Her loves he.
   b. *It seems Brett to like beer.
However, there are some things which cannot be ruled out by these two modules:

(3) a. *John love her.  
b. *I loves him.

- Verbs must agree with subjects.  
- In English, this pattern is poor:

<table>
<thead>
<tr>
<th></th>
<th>Singular</th>
<th>Plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>l speak</td>
<td>we speak</td>
</tr>
<tr>
<td>2nd</td>
<td>you speak</td>
<td>you speak</td>
</tr>
<tr>
<td>3rd</td>
<td>s/he speak-s</td>
<td>they speak</td>
</tr>
</tbody>
</table>
Agreement

• Verbs must agree with subjects.
• In Italian, this pattern is more complicated:

<table>
<thead>
<tr>
<th></th>
<th>Singular</th>
<th>Plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>io parl-o</td>
<td>noi parl-iamo</td>
</tr>
<tr>
<td>2nd</td>
<td>tu parl-i</td>
<td>voi parl-ate</td>
</tr>
<tr>
<td>3rd</td>
<td>lui/lei parl-a</td>
<td>loro parl-ano</td>
</tr>
</tbody>
</table>

• There is a syntactic dependency between the subject and the verb.
Agreement is meaningless

The agreement suffixes on the verb are meaningless:

- Some English dialects use agreement differently, but mean exact same thing as Standard English speakers:
  
  (4) John love-∅ Mary. (African-American English)
  (5) I walk-s. (West Country English, Somerset, Dorset)

- English speakers can easily understand non-native speakers with agreement mistakes:
  
  (6) I walks. (= I walk., ≠ He walks.)
Shared features

(7) John plays piano.

\[ \text{3.sg} \quad \text{3.sg} \]

- Both the subject and the verb have \textit{3rd.singular} feature on them.
- On the \textit{subject}, this feature is \textit{meaningful}: it indicates that John is a singular 3rd person entity.
  - \textit{Interpretable feature}: \langle i3sg \rangle
- On the \textit{verb}, this feature is \textit{meaningless}, and occurs as a result of dependency with the subject. We completely ignore it when interpreting the sentence from semantic point of view.
  - \textit{Uninterpretable feature}: \langle u3sg \rangle
Interpretability of features

(8) *I plays piano.

\langle i1.sg \rangle \langle u3.sg \rangle

There is mismatch of features: \langle i1.sg \rangle and \langle u3.sg \rangle

For each uninterpretable feature, there should be a similar interpretable feature. Basically, we don’t want to have something that receives no interpretation!

Principle of Interpretability

Any clause in which some element carries an uninterpretable feature \langle uF \rangle requires the presence of a matching interpretable feature \langle iF \rangle; otherwise the clause is ungrammatical.
Formal Theory of Features
Let us explore what kind of features the grammar uses.

T-feature

- T carries Tense-features (T-features), unless it’s non-finite.
- T-feature on T is interpretable: depending on the value, it situated the event on the time line: \( \langle iT: \_\_ \rangle \)
- Non-finite T to has no specification for T-features.
- In English, only present and past are marked on the verb. The future tense is expressed using a modal will.
- T-feature on Verb itself is uninterpretable T-feature, but it comes with a value, which must be transmitted to T: \( \langle uT: \text{present/past} \rangle \).
Agreement is also a feature on T.

- It is clear if we have **auxiliary verbs**: usually they agree with the subject.
- Otherwise, think of **Affix hopping**: Tense is on T, and it determines which affix the verb has. **Verbal affixes depend on T!**
- Agreement features (person/number) are called **φ-features**.
  - φ-features are interpretable on DP: for example, \( i\varphi: 3.\text{sg} \)
  - φ-features are non-interpretable on T: \( u\varphi: \_\_ \)
- φ-features are specified similar to T-features, for example, on DP *John* we have \( i\varphi: 3.\text{sg} \).
Feature checking: Feature with no value (unvalued) searches for a matching feature below, and acquires its value from it.

- For example, T-feature on T is unvalued, so it will look down, find a V with a valued T-feature, and get its value from it.
**Feature checking:** Feature with no value (unvalued) searches for a matching feature below, and acquires its value from it.

- For example, \( \varphi \)-feature on \( T \) is unvalued, so it will look down, find a subject DP with a valued \( \varphi \)-feature, and get its value from it.

```
TP
  / \  
DP_i  T'  
    /   
   T     VP
     /   
    DP_i V' 
      /   
     V    DP
```

- **Feature Checking**
- **Subject Movement**
(9)  John loves me: Tree before feature checking.
Example: Feature distribution

(10) John loves me: Feature checking before subject movement to Spec, TP.
Example: Feature distribution

(11) John loves me: Final tree, values have been copied to T.

\[\langle i\varphi: 3.\text{sg}\rangle \rightarrow\langle iT: \text{pres}\rangle \rightarrow\langle u\varphi: 3.\text{sg}\rangle \rightarrow EPP \rightarrow \langle i\varphi: 3.\text{sg}\rangle \rightarrow T \rightarrow \langle uT: \text{pres}\rangle \rightarrow \langle i\varphi: 1.\text{sg}\rangle \rightarrow \text{me} \rightarrow \text{loves} \rightarrow \text{John} \rightarrow \text{DP}_i \rightarrow \text{VP} \rightarrow \text{T'} \rightarrow \text{TP} \rightarrow \text{DP}_i \rightarrow \text{John} \rightarrow \text{TP} \rightarrow \text{TP} \]
Subjects agree, but not objects

(12) John loves me: Tree before subject movement to Spec, TP

- T finds subject first, so its $\varphi$-features are valued from the subject. That’s why subjects agree, and not objects.

```
T’
  /\ T
 /   VP
\    /
\    DP_i
\   /
\   John
\   /
\   /
\   \ 3.sg
\   v
\   \ 1.sg
\   V’
\   |
\   V
\   /
\   loves
\   /
\   /
\   \ pres
\   x
\   |
\   DP
\   me
```
Nominative case

We saw that nominative case is only available to subjects of the finite clauses:

(13) a. John loves me.
    b. *[John to love me] would be amazing.
    c. [For John to love me] would be amazing.

How can this be captured in terms of features?

- **Idea:** Nominative case is an uninterpretable T-feature on DP.
- This idea belongs to Pesetsky & Torrego (2007).
- Before there was an independent ⟨Case⟩-feature...
Case as a T-feature

(14) He loves John: Feature checking before subject movement to Spec, TP

- T first enters Agree with the subject, and then with the Verb. As a result, both the T and the subject will get their T-features from V.
Case as a T-feature

(15) He loves John: After feature checking; values are copied.

- After copying values, the subject has a **valued T-feature**, which is pronounced as a **nominative case**.

\[
\begin{array}{c}
\text{T'} \\
\text{VP} \\
\text{DP_i} \\
\text{he} \\
\text{V'} \\
\text{loves} \\
\text{DP} \\
\text{John}
\end{array}
\]

\[
\begin{array}{c}
\langle iT: \text{pres}\rangle \\
\langle u\varphi: 3.sg\rangle \\
\varphi \\
\langle i\varphi: 3.sg\rangle \\
\langle uT: \text{pres}\rangle \\
\langle uT: \text{pres}\rangle \\
\langle i\varphi: 3.sg\rangle
\end{array}
\]
### Feature distribution summary

#### Main syntactic features

<table>
<thead>
<tr>
<th>Element</th>
<th>Features</th>
<th>Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>$\varphi$-features</td>
<td>interpretable, valued</td>
<td>$\langle i\varphi: 3.\text{sg} \rangle$</td>
</tr>
<tr>
<td></td>
<td>T-features/Case</td>
<td>uninterpretable, unvalued</td>
<td>$\langle iT: __ \rangle$</td>
</tr>
<tr>
<td>V</td>
<td>T-features</td>
<td>uninterpretable, valued</td>
<td>$\langle iT: \text{pres} \rangle$</td>
</tr>
<tr>
<td>T</td>
<td>T-features/Case</td>
<td>interpretable, unvalued</td>
<td>$\langle iT: __ \rangle$</td>
</tr>
<tr>
<td></td>
<td>$\varphi$-features</td>
<td>uninterpretable, unvalued</td>
<td>$\langle u\varphi: __ \rangle$</td>
</tr>
</tbody>
</table>
Searching for a matching feature

**Probe-Goal relationship**

- **Probe**: the head with a feature which is searching.
- **Goal**: whatever is being searched for.

For example, with respect to $\varphi$-features, $T$ is a probe, and the subject $DP$ is a goal.

One question left to answer is:

- How do the unvalued features search for valued features?
- Where do Probes look for their Goals?
- To answer this, we need to introduce an important relation between the nodes in the tree: $c$-command.