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

θ-Theory

There are two modules which we so far considered which limit X-bar theory:

1. θ-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

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.

Agreement

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:

	Singular		Plural	
1st	I	speak	we	speak
2nd	you	speak	you	speak
3rd	s/he	speak-s	they	speak

Agreement

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

	Singular		Plural	
1st	io	parl-o	noi	parl-iamo
2nd	tu	parl-i	voi	parl-ate
3rd	lui/lei	parl-a	loro	parl-ano

- 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.
3.SG 3.SG

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

Interpretability of features

(8) *I plays piano.
⟨*i*1.sg⟩ ⟨*u*3.sg⟩

There is mismatch of features: ⟨*i*1.sg⟩ and ⟨*u*3.sg⟩
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 ⟨*u*F⟩ requires the presence of a matching interpretable feature ⟨*i*F⟩; otherwise the clause is ungrammatical.

Formal Theory of Features

Tense-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: ⟨*i*T: ___⟩
- 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: ⟨*u*T: present/past⟩.

φ-features

φ-features

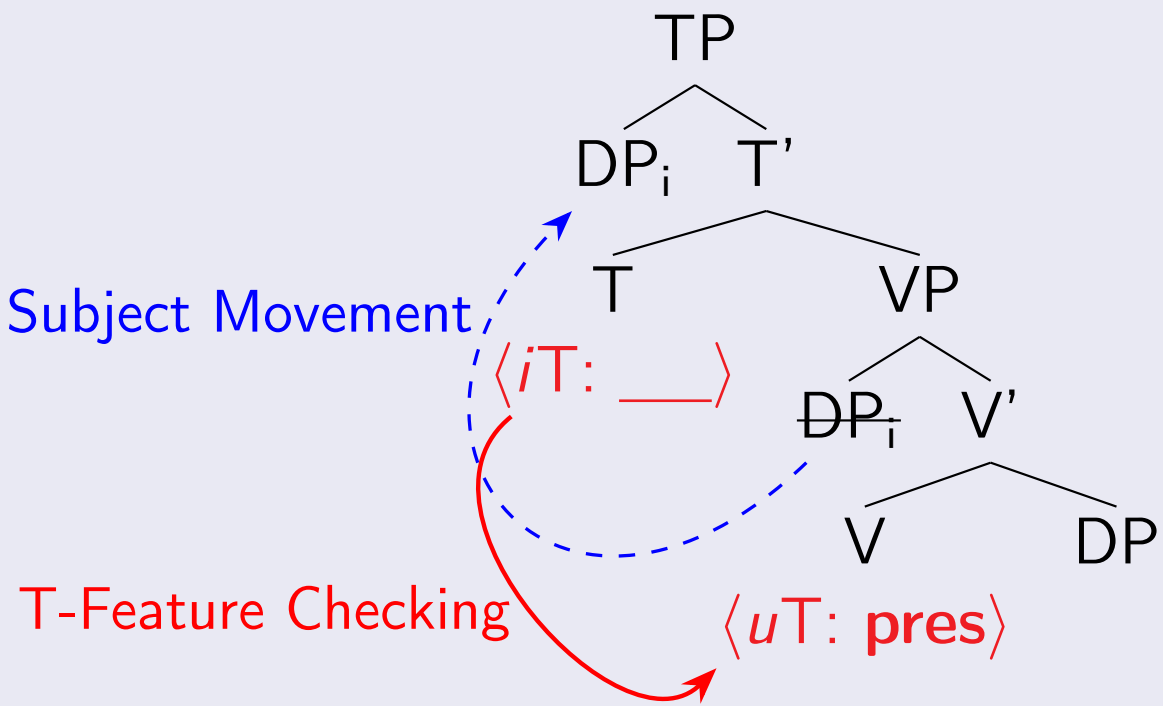
Agreement is also a feature on T.

- 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, $\langle i\varphi$: 3.sg$\rangle$
  - φ-features are non-interpretable on T: $\langle u\varphi$: ___$\rangle$
- φ-features are specified similar to T-features, for example, on DP *John* we have $\langle i\varphi$: 3.sg$\rangle$.

Feature checking

Feature checking: Feature with no value (unvalued) searches for a matching feature below, and acquires its value from it.

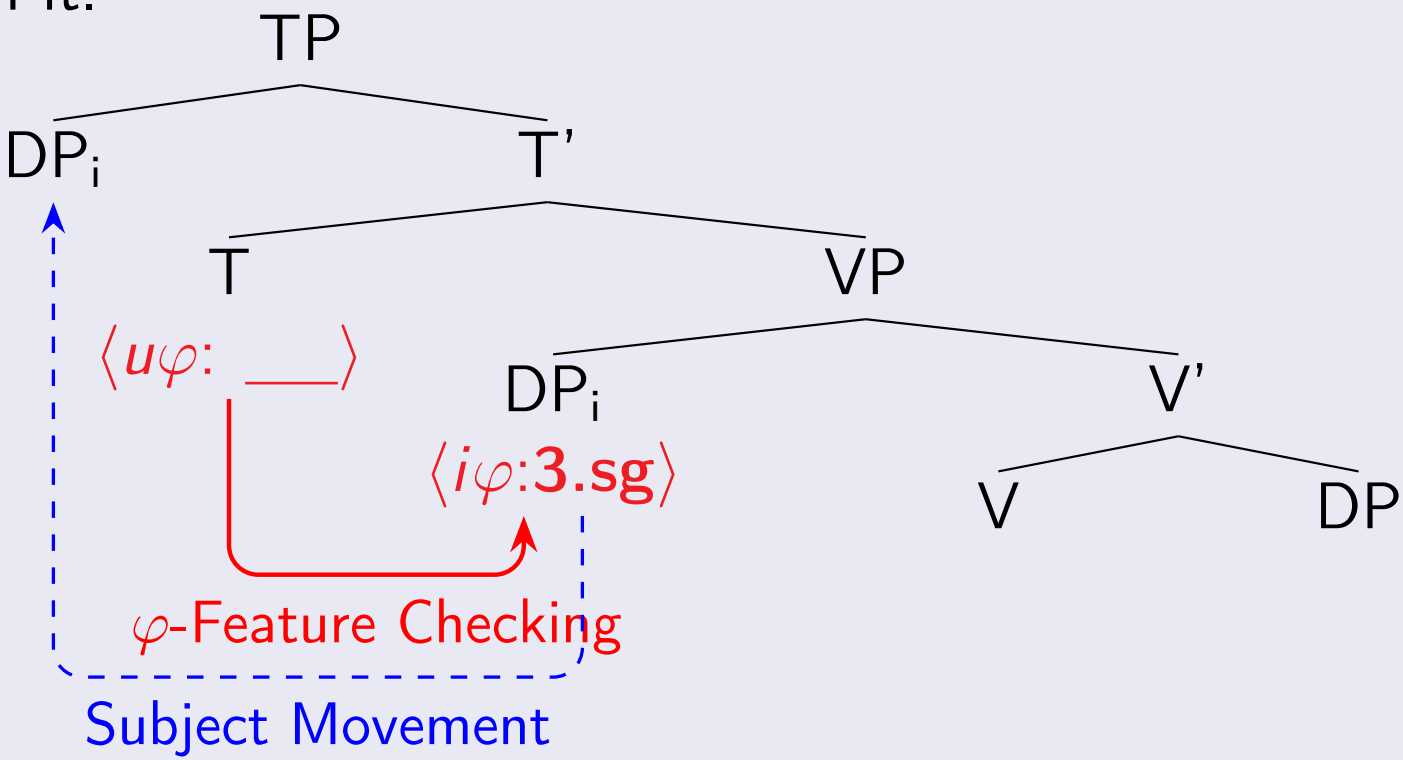
- For example, T-feature on T is unvalued, so it will look down, find a V with a valued T-feature, and get its value from it.



Feature checking

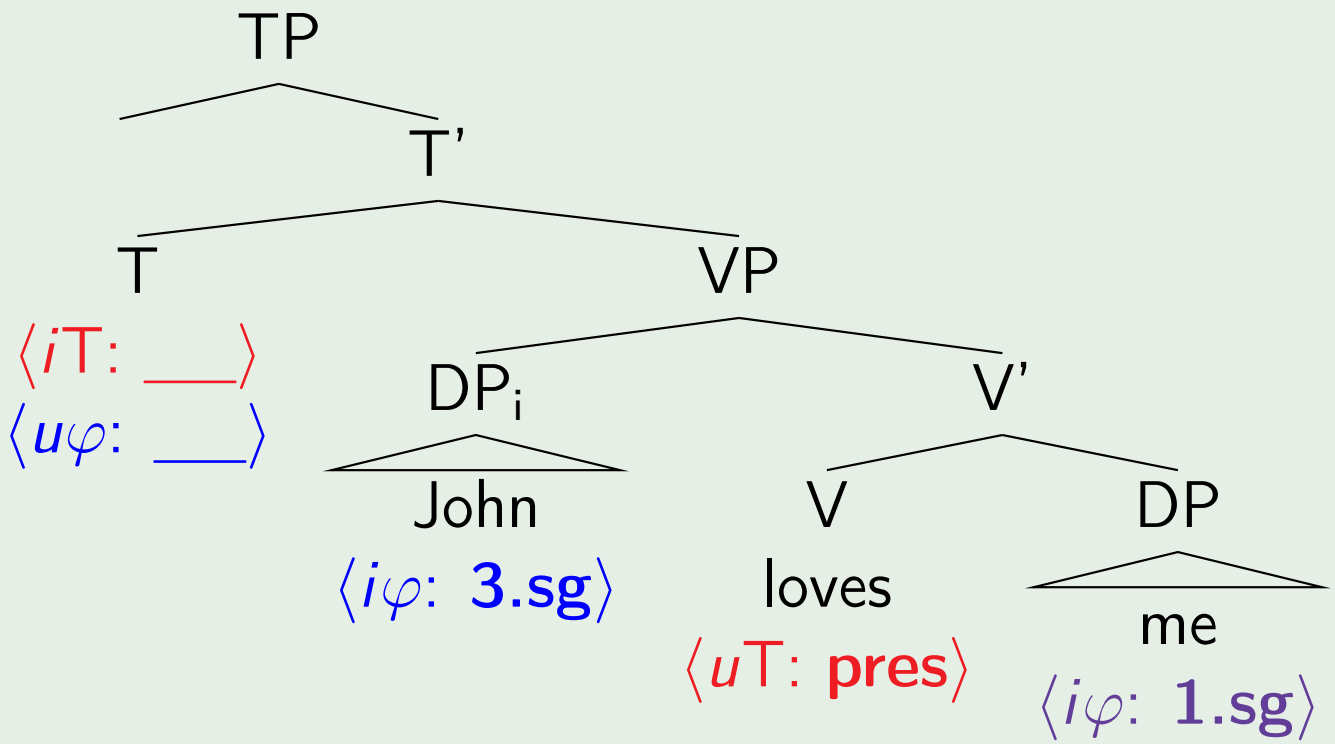
Feature checking: Feature with no value (unvalued) searches for a matching feature below, and acquires its value from it.

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



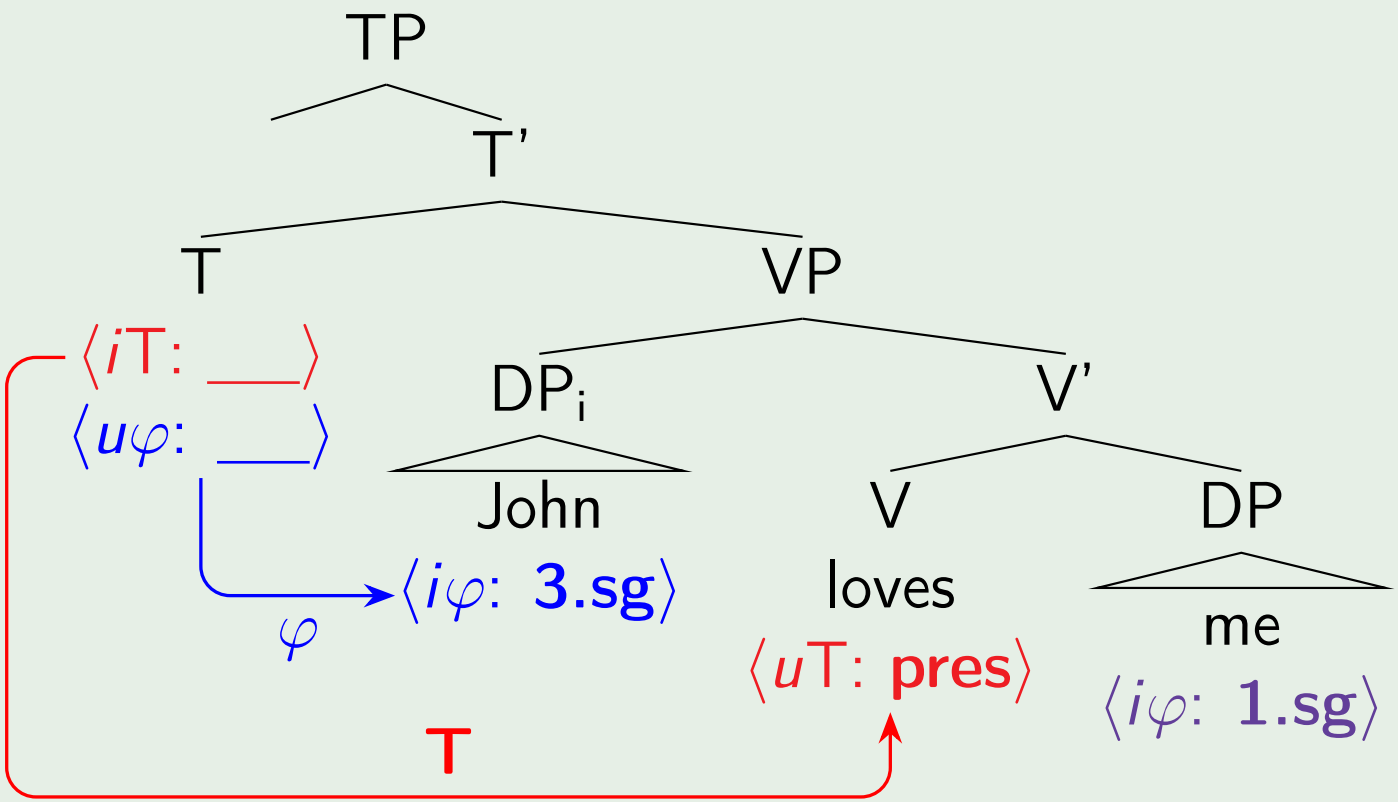
Example: Feature distribution

(9) John loves me: Tree before feature checking.



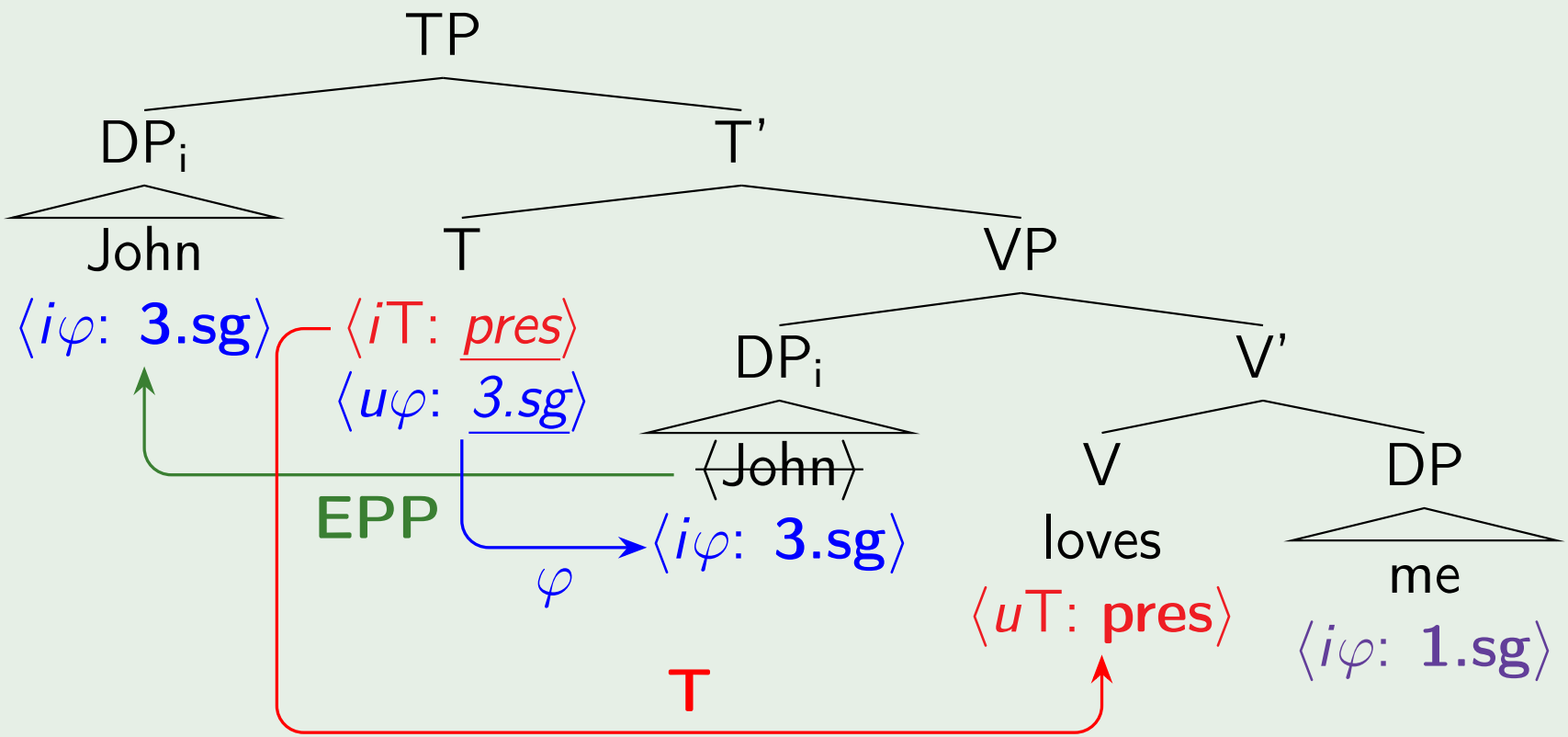
Example: Feature distribution

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



Example: Feature distribution

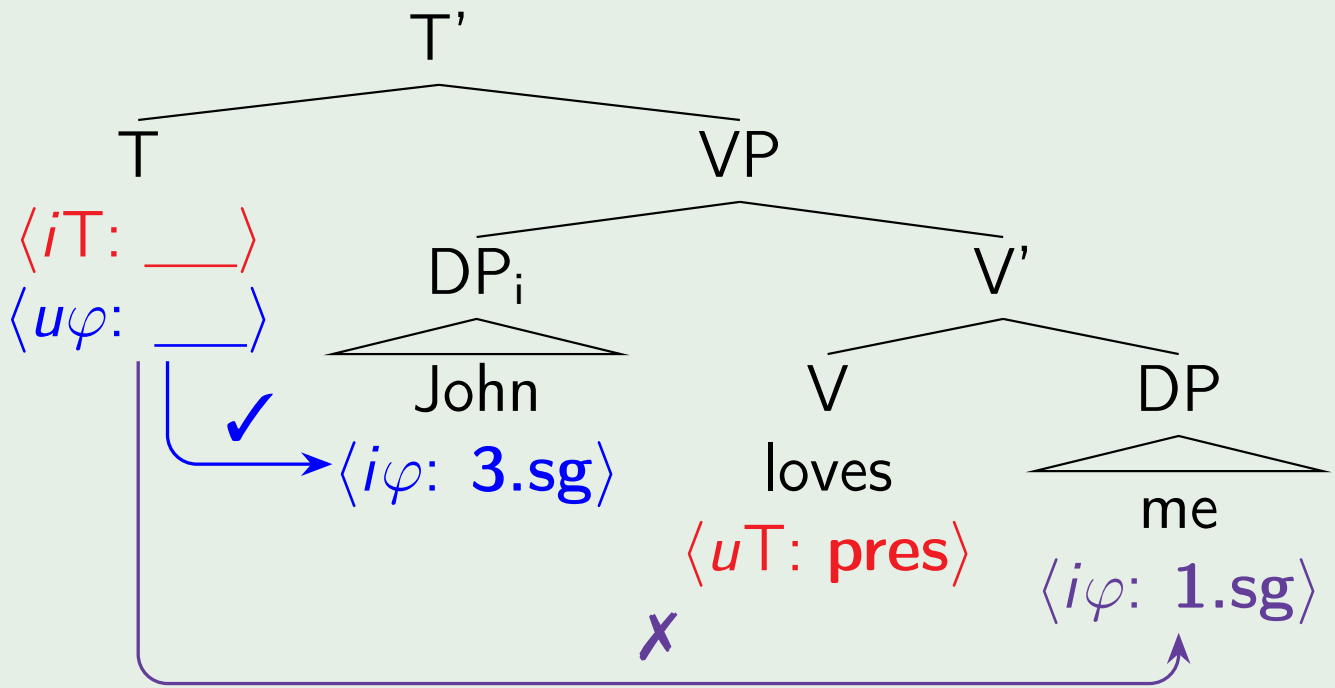
(11) John loves me: Final tree, values have been copied to T.



Subjects agree, but not objects

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

- T finds subject first, so its φ-features are valued from the subject. That's why subjects agree, and not objects.



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.

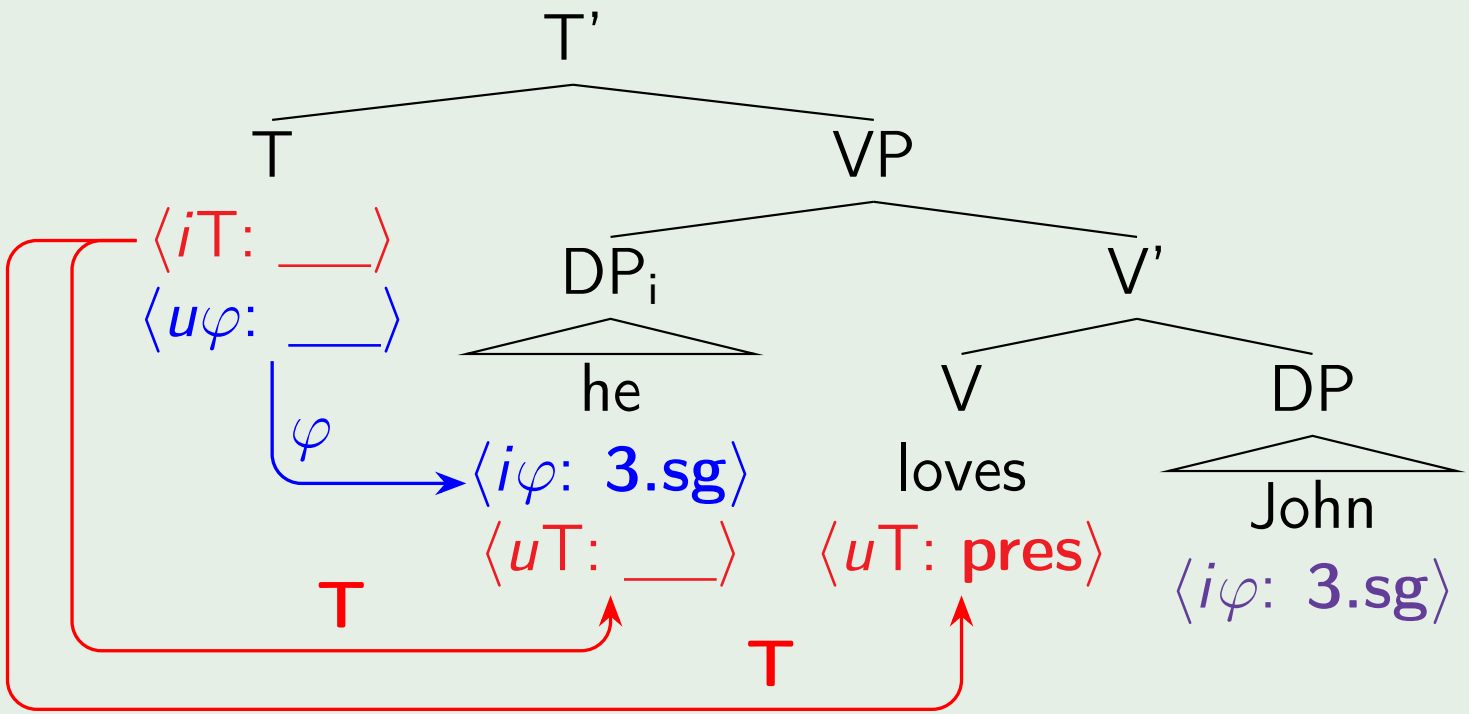
How can this be captured in terms of features?

- **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

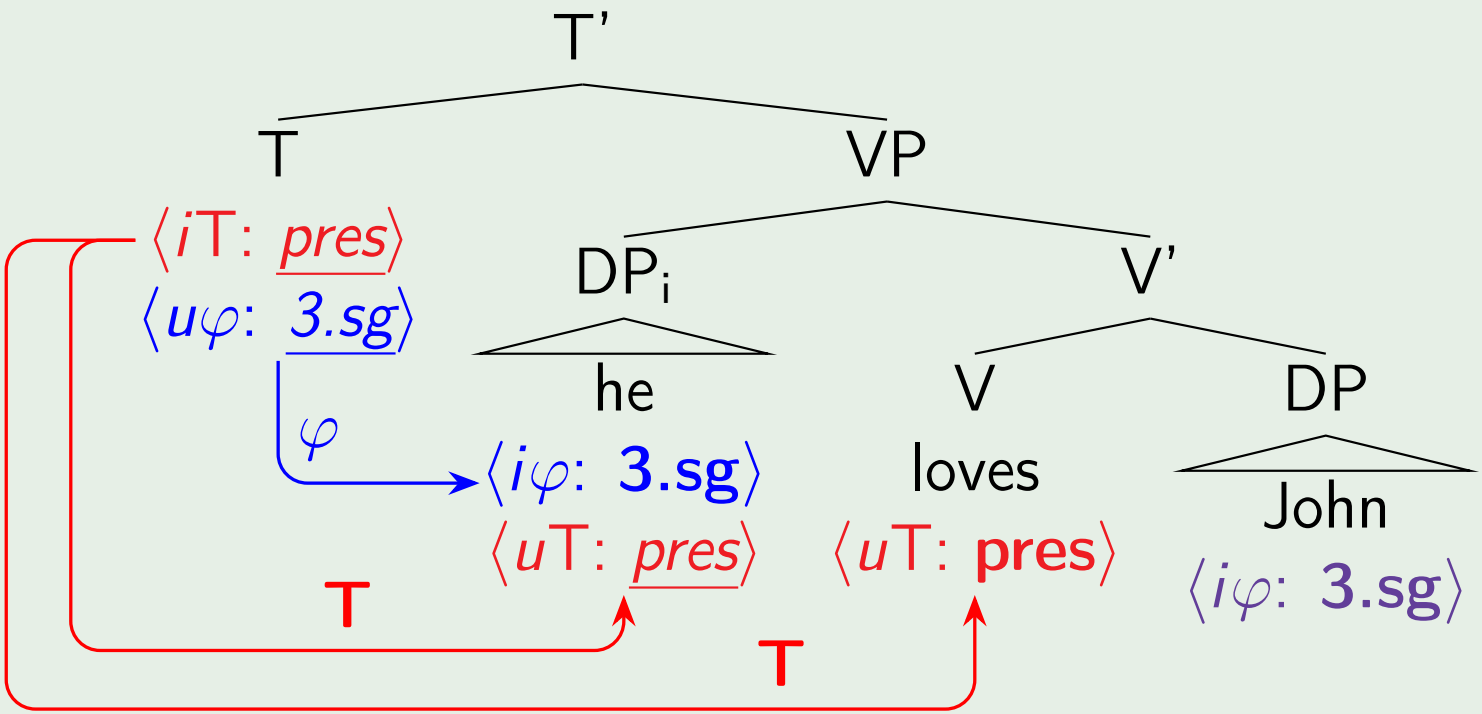
- T first enters Agree with the subject, and then with the Verb. As a result, both the T and the subject will get their T-features from V.



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.



Feature distribution summary

Main syntactic features

Element	*Features*	*Type*	*Example*
DP	φ-features	interpretable, valued	$\langle i\varphi$: **3.sg**$\rangle$
	T-features/Case	uninterpretable, unvalued	$\langle u$T: ___$\rangle$
V	T-features	uninterpretable, valued	$\langle i$T: **pres**$\rangle$
T	T-features/Case	interpretable, unvalued	$\langle i$T: ___$\rangle$
	φ-features	uninterpretable, unvalued	$\langle u\varphi$: ___$\rangle$

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 φ-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.