

Wh-movement

movement of question phrases

Another Terminological Point



Another Terminological Point

- A Movement is the same thing as DP movement
 - Subtypes: Passive, SSR, SOR, movement of subject from VP internal subject to spec, TP

Another Terminological Point

- A Movement is the same thing as DP movement
 - Subtypes: Passive, SSR, SOR, movement of subject from VP internal subject to spec, TP
- A' (A-bar) Movement is the same thing as Wh-movement
 - Subtypes: Wh-movement, topicalization, relative clause formation

Two kinds of Questions



Two kinds of Questions

- Yes/No questions:

- Did you see the octopus?

yes/no /*dog

- Have you eaten yet?

yes/no /*apple

Two kinds of Questions

- Yes/No questions:

- Did you see the octopus?

yes/no /*dog

- Have you eaten yet?

yes/no /*apple

- Wh-questions

- Who was here last week?

Howard/*no

- What do you have there?

Nail clippers/*yes

Wh words

who
what
which X

Usually arguments

whose X

where
when
why
how

Usually adjuncts

Wh words

who
what
which X

Usually arguments

whose X

Can be either adjunct or argument

where
when
why
how

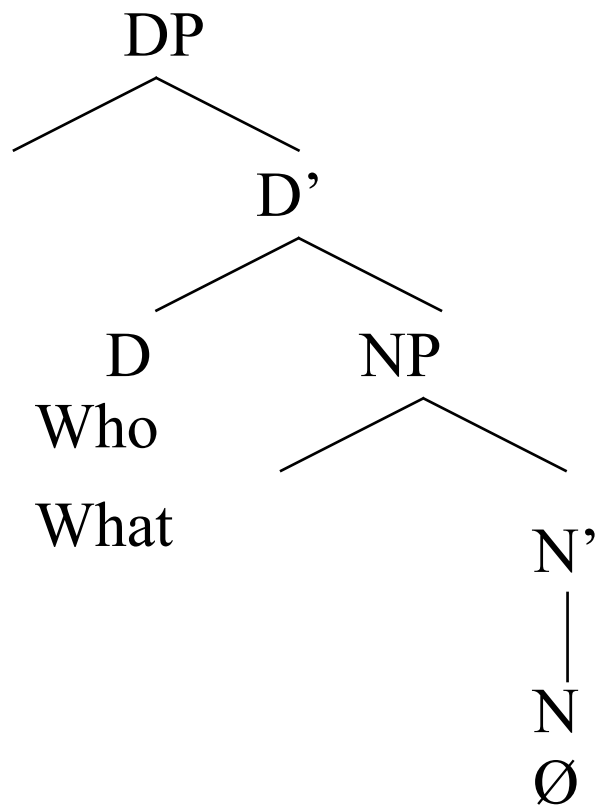
Usually adjuncts

Wh-phrases are what move

- We move *wh-phrases*,
- *Who, What, Which, Whose* are determiners:

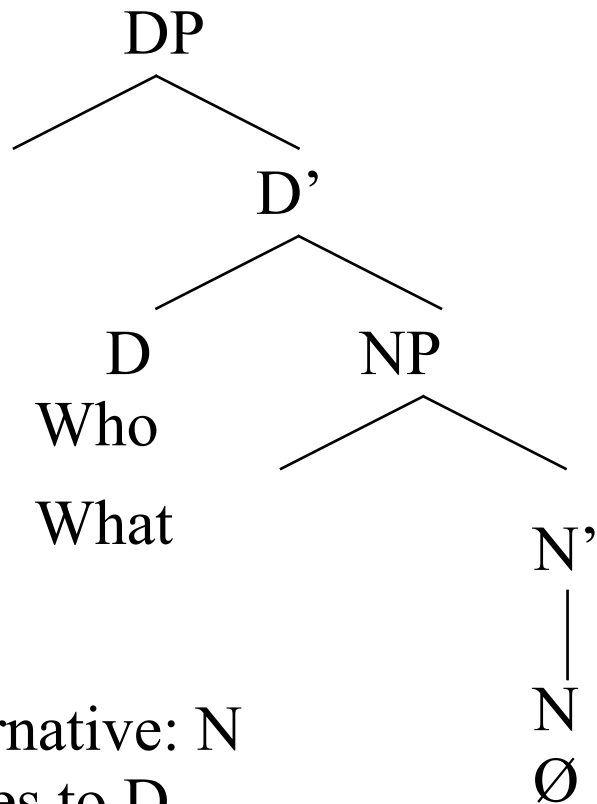
Wh-phrases are what move

- We move *wh-phrases*,
- *Who, What, Which, Whose* are determiners:



Wh-phrases are what move

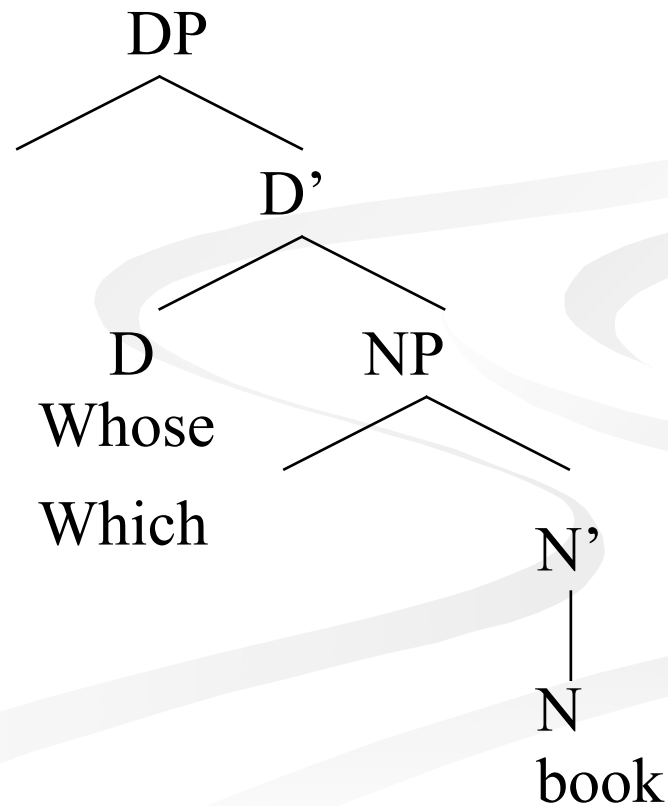
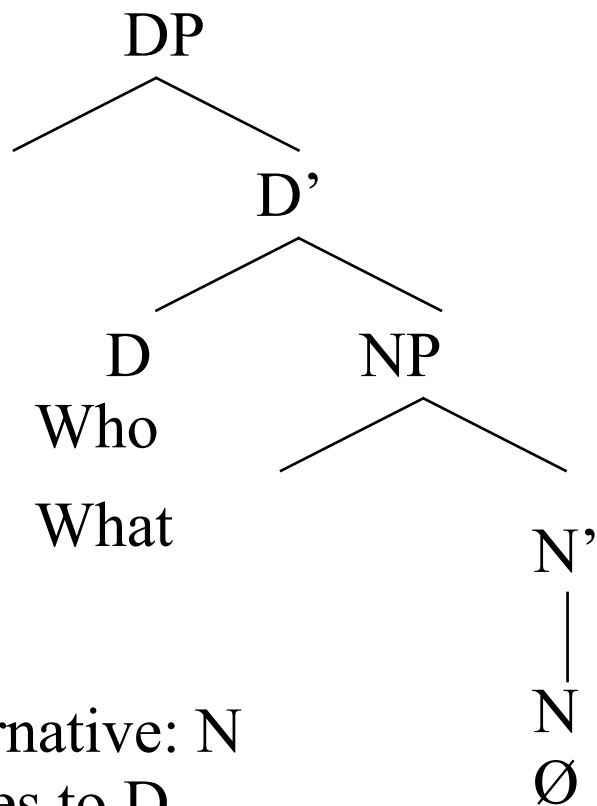
- We move *wh-phrases*,
- *Who, What, Which, Whose* are determiners:



Alternative: N
moves to D

Wh-phrases are what move

- We move *wh-phrases*,
- *Who, What, Which, Whose* are determiners:



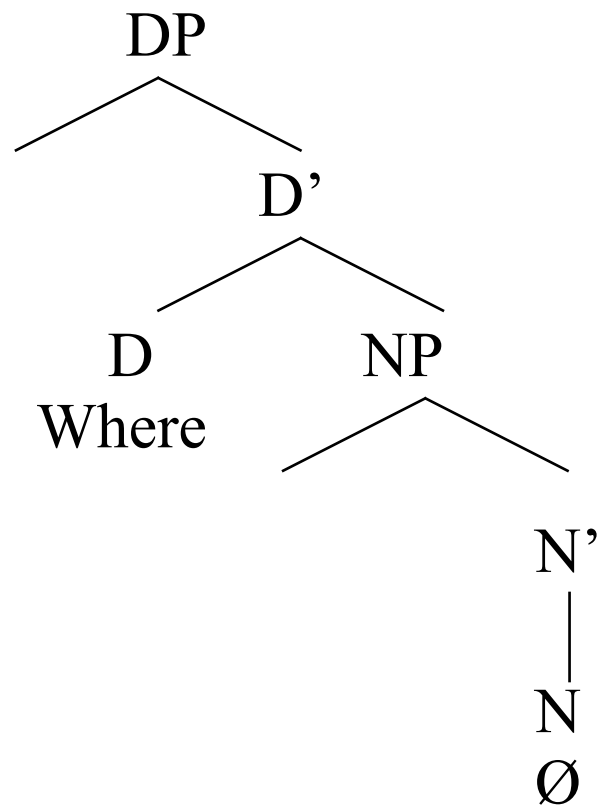
Alternative: N
moves to D

Wh-phrases are what move

- *Where* is sometimes a determiner
 - Where did John go to (cf. John went to school)
- And sometimes an Adverb
 - Where did John go (cf. John went home)

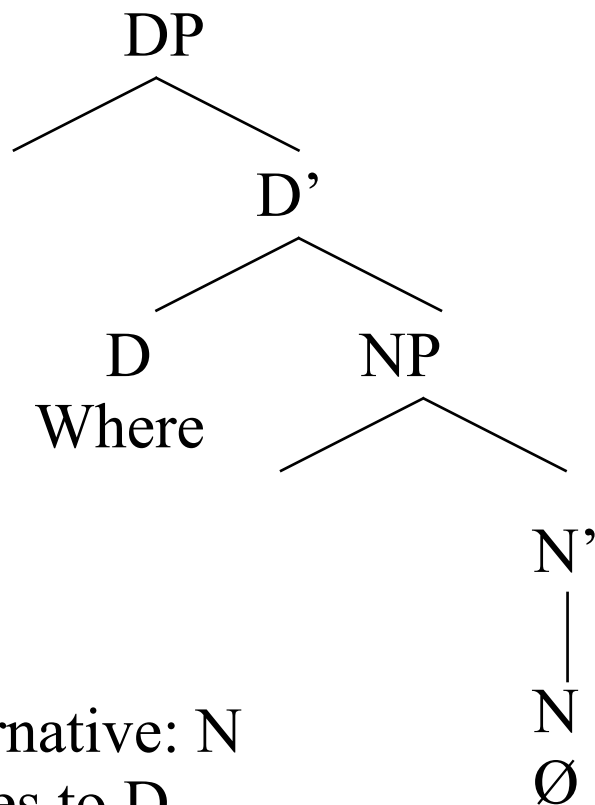
Wh-phrases are what move

- *Where* is sometimes a determiner
 - Where did John go to (cf. John went to school)
- And sometimes an Adverb
 - Where did John go (cf. John went home)



Wh-phrases are what move

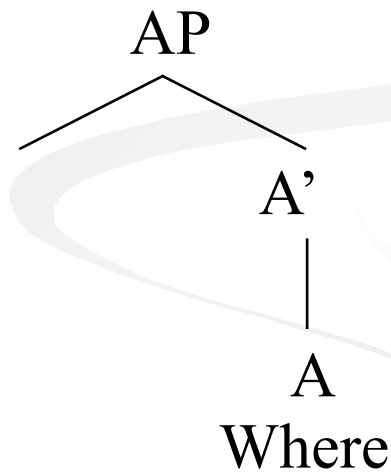
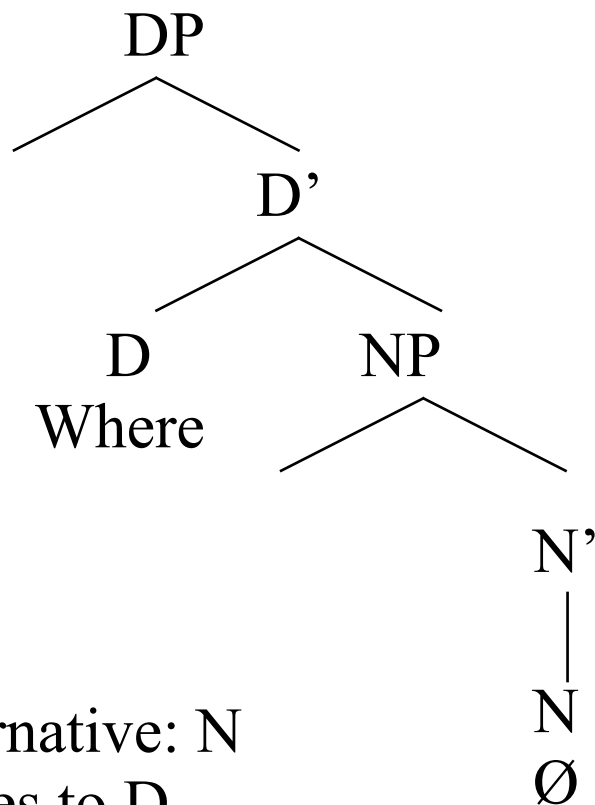
- *Where* is sometimes a determiner
 - Where did John go to (cf. John went to school)
- And sometimes an Adverb
 - Where did John go (cf. John went home)



Alternative: N
moves to D

Wh-phrases are what move

- *Where* is sometimes a determiner
 - Where did John go to (cf. John went to school)
- And sometimes an Adverb
 - Where did John go (cf. John went home)



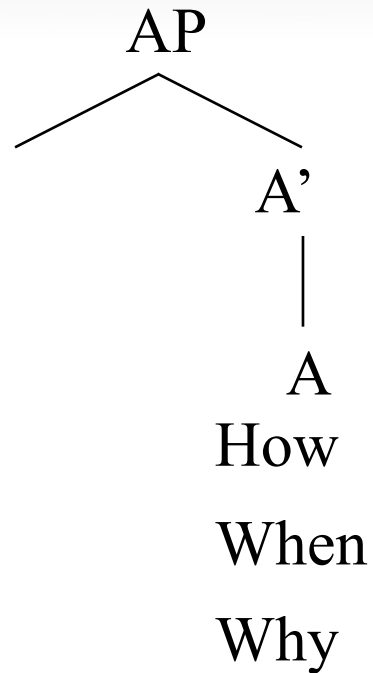
Alternative: N
moves to D
©Andrew Carnie, 2006

Wh-phrases are what move

- *How, When, Why* are usually adverbs

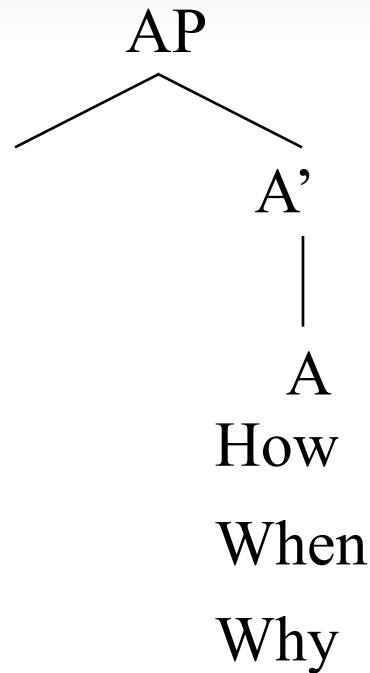
Wh-phrases are what move

- *How, When, Why* are usually adverbs



Wh-phrases are what move

- *How, When, Why* are usually adverbs



REMEMBER:
The *whole* phrase
moves, not just the head

Wh-questions involve movement

- I bought a book
- What did you buy _____
↑



Where from?

What did you say [_____ was hit _____]

Where from?

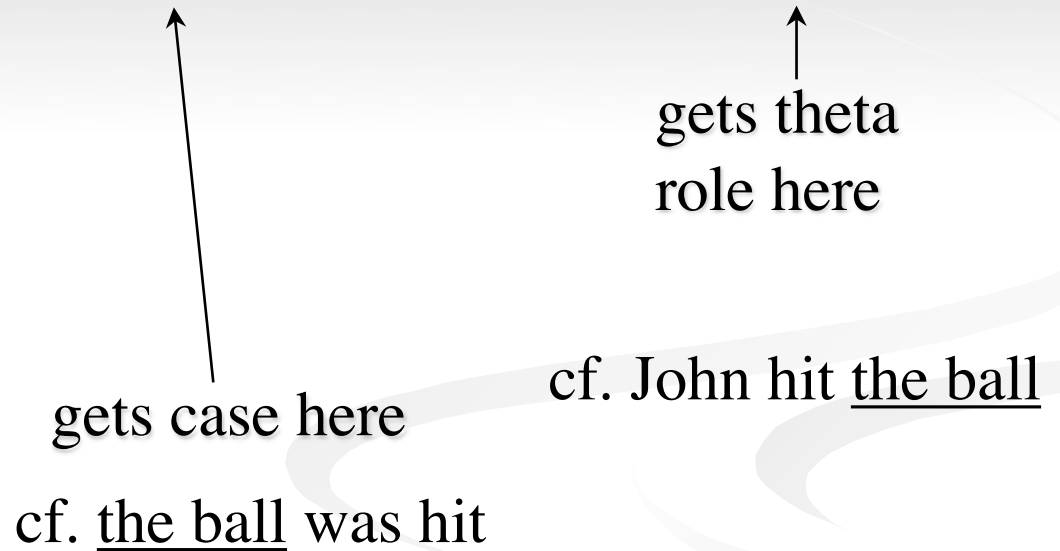
What did you say [_____ was hit _____]

↑
gets theta
role here

cf. John hit the ball

Where from?

What did you say [_____ was hit _____]



Where from?

What did you say [_____ was hit _____]

↑
gets theta
role here

cf. John hit the ball

↑
gets case here

cf. the ball was hit

↑
Ends up here

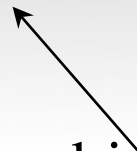
Where to?

What have you seen _____



Where to?

What have you seen _____



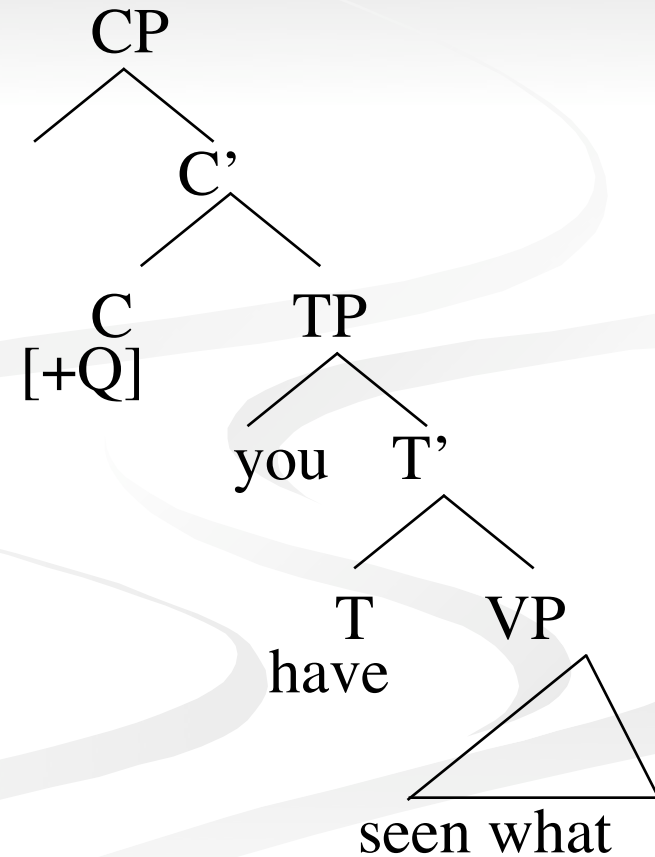
subject aux inversion:

means Aux is in C

Where to?

What have you seen _____

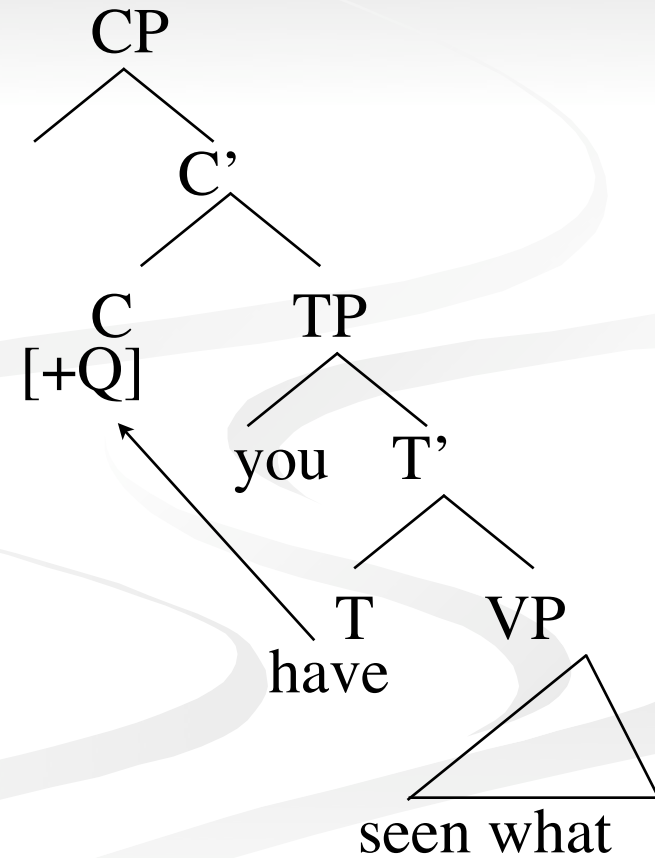
subject aux inversion:
means Aux is in C



Where to?

What have you seen _____

subject aux inversion:
means Aux is in C

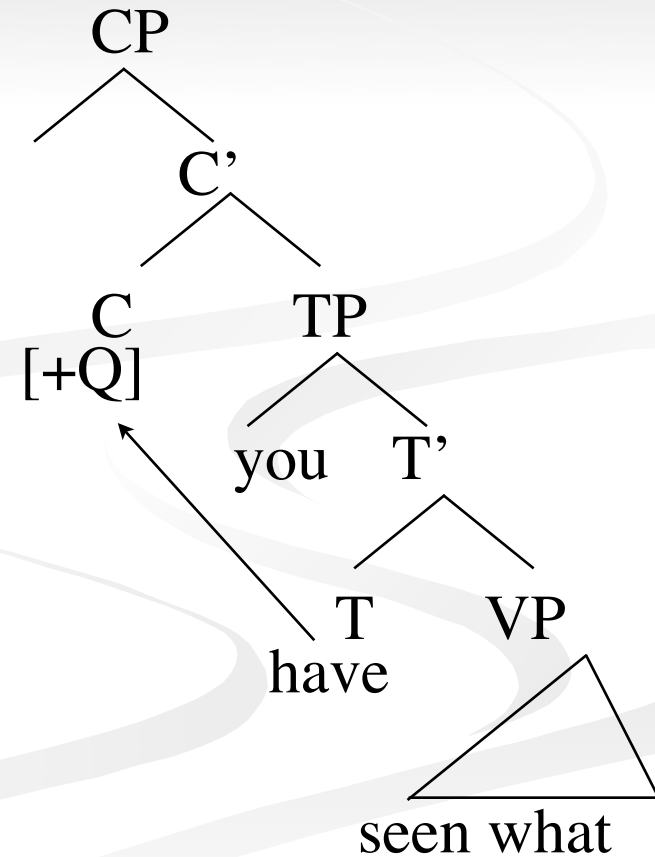


Where to?

What have you seen _____

subject aux inversion:
means Aux is in C

Wh-word precedes C --
specifier of CP

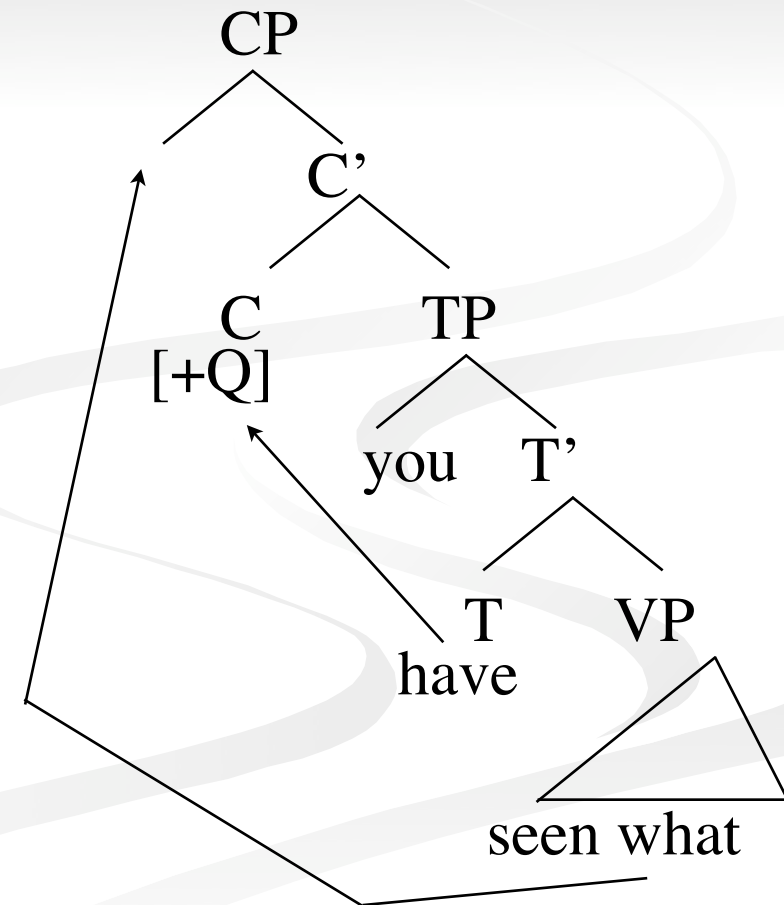


Where to?

What have you seen _____

subject aux inversion:
means Aux is in C

Wh-word precedes C --
specifier of CP



Why?



Why?

Movement of T to C is motivated by [+Q]

Why?

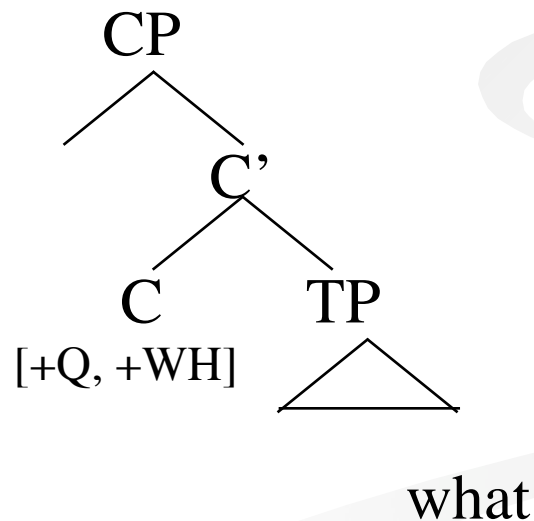
Movement of T to C is motivated by [+Q]

Proposal: there is a [+WH] feature in C, the *wh*-word must get close to it.

Why?

Movement of T to C is motivated by [+Q]

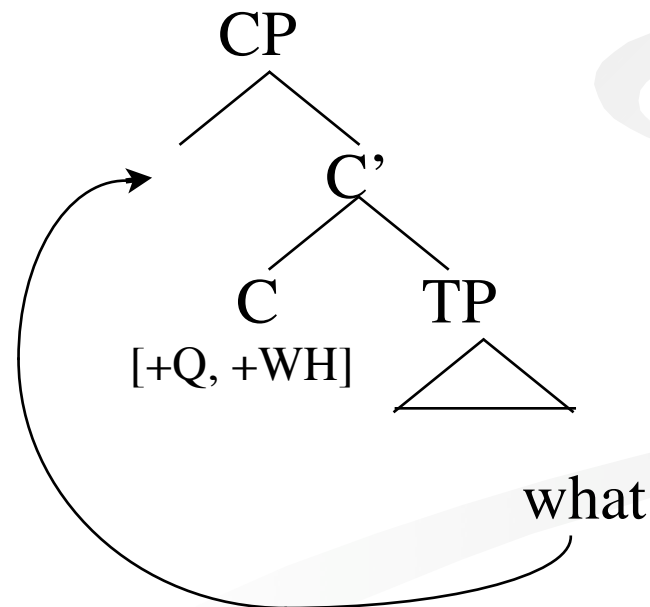
Proposal: there is a [+WH] feature in C, the *wh*-word must get close to it.



Why?

Movement of T to C is motivated by [+Q]

Proposal: there is a [+WH] feature in C, the *wh*-word must get close to it.



[+WH] Complementizers

Cé a^L bhí sa seomra?

Who that-wh was in-the room

“Who was in the room”

Motivations for movements

- *Head Movement:*

- V to T (T to V) motivated by need of suffix
- T to C motivated by null [+Q] C

- *NP movement*

- Raising motivated by need for case
- Passive motivated by need for case

- *Wh movement*

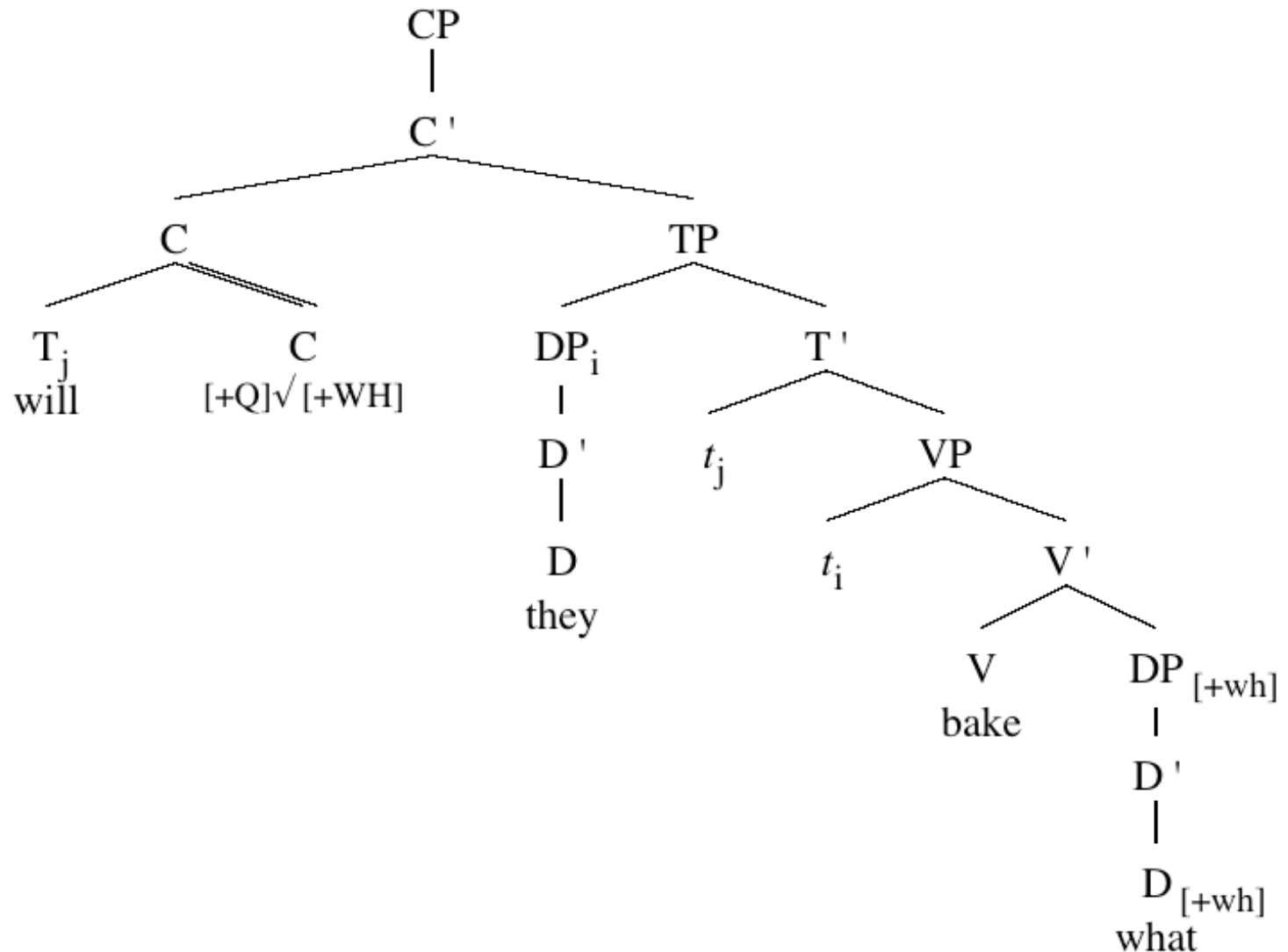
- *wh*-questions motivated by need for *wh*-word to appear near [+*wh*] complementizer.

Wh-questions

1 thing to solve:

- [+Q] C needs T
- [+WH] C needs a [+wh] spec.

Head-movement of *will* to C will check the Q feature

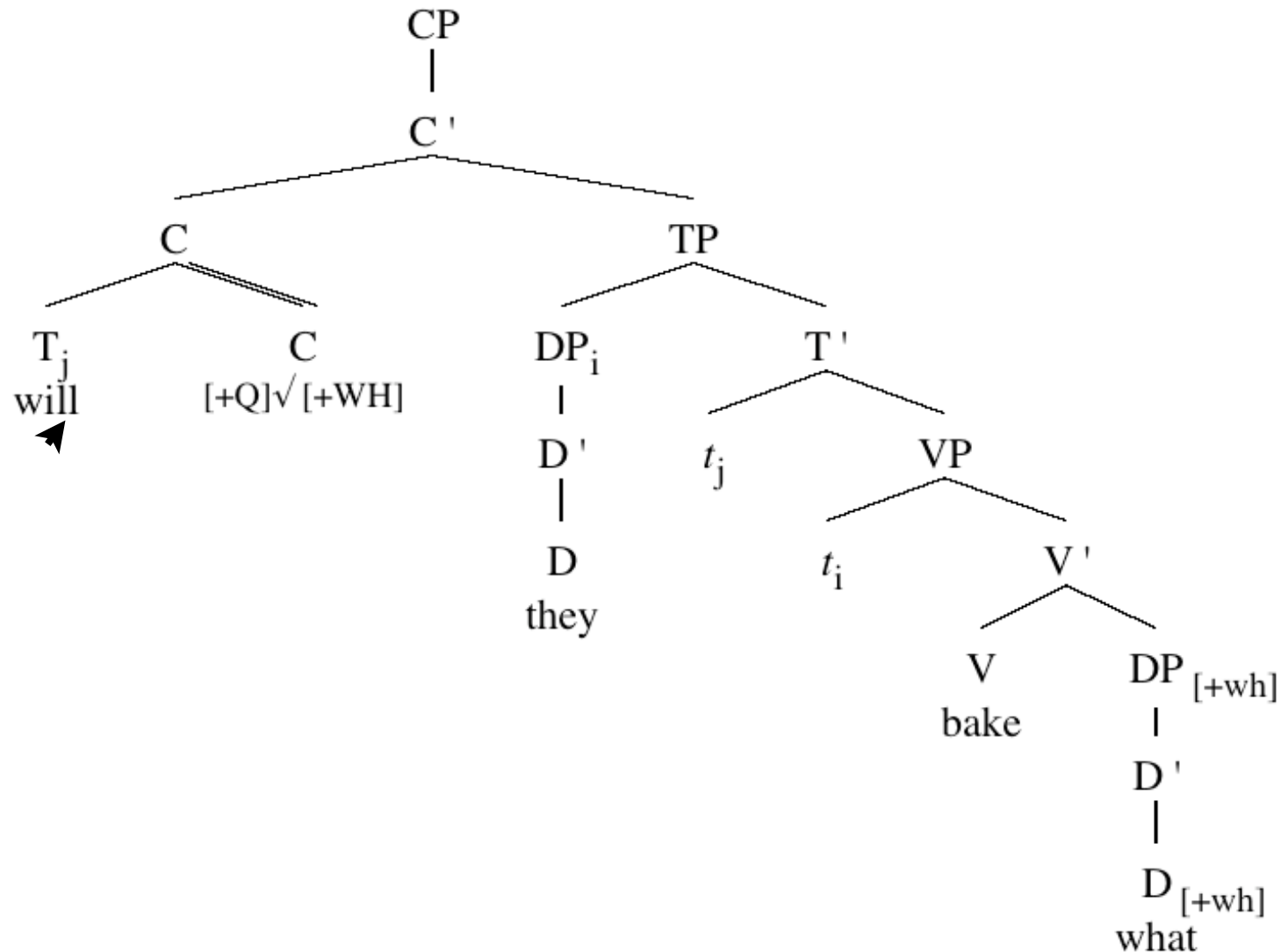


Wh-questions

1 thing to solve:

- [+Q] C needs T
- [+WH] C needs a [+wh] spec.

Head-movement of *will* to C will check the Q feature



Two weird English-specific constraints

- *Who that John left?
- *John asked who if Susan loved?
- English doesn't allow you to have both an overt complementizer (other than Aux) and a wh-word
- *The Doubly filled CP filter* (English only)
 - * [_{CP} wh that]

Two weird English-specific constraints

- Who did John think that Susan loved ____?
- *Who did John think that ____ loved Susan?
- Who did John think \emptyset ____ loved Susan?
- can't wh-move from a position next to the word "that".
- *That-trace filter* (English only)
 - * that *t*

A Derivation

What did John say was baked?



A Derivation

What did John say was baked?

Step 1. UNDO all the transformations to figure out the D-structure

A Derivation

What did John say was baked?

Step 1. UNDO all the transformations to figure out the D-structure

- 1) T→C movement: What John did say was baked
- 2) Do insertion: What John -ed say was baked

A Derivation

What did John say was baked?

Step 1. UNDO all the transformations to figure out the D-structure

- 1) T→C movement: What John did say was baked
- 2) Do insertion: What John -ed say was baked

Question: where did “What” start?

A Derivation

What did John say was baked?

Step 1. UNDO all the transformations to figure out the D-structure

- 1) T→C movement: What John did say was baked
- 2) Do insertion: What John -ed say was baked

Question: where did “What” start?

Notice: John said the cake was baked (passive in embedded clause)

A Derivation

What did John say was baked?

Step 1. UNDO all the transformations to figure out the D-structure

- 1) T→C movement: What John did say was baked
- 2) Do insertion: What John -ed say was baked

Question: where did “What” start?

Notice: John said the cake was baked (passive in embedded clause)

- 3) Wh-movement: John -ed say what was baked

A Derivation

What did John say was baked?

Step 1. UNDO all the transformations to figure out the D-structure

- 1) T→C movement: What John did say was baked
- 2) Do insertion: What John -ed say was baked

Question: where did “What” start?

Notice: John said the cake was baked (passive in embedded clause)

- 3) Wh-movement: John -ed say what was baked
- 4) DP-movement: John -ed say was baked what

A Derivation

What did John say was baked?

Step 1. UNDO all the transformations to figure out the D-structure

- 1) T→C movement: What John did say was baked
- 2) Do insertion: What John -ed say was baked

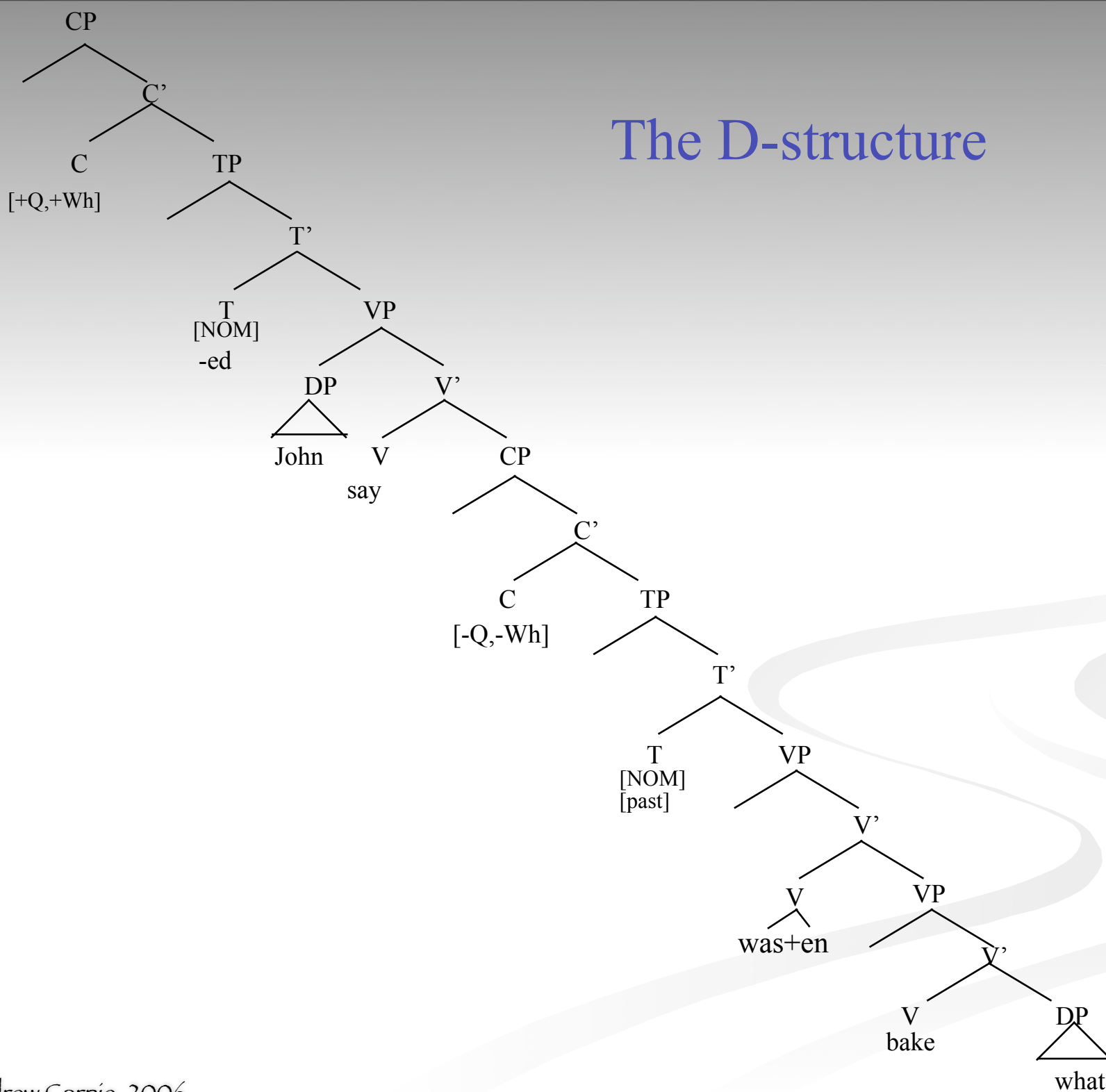
Question: where did “What” start?

Notice: John said the cake was baked (passive in embedded clause)

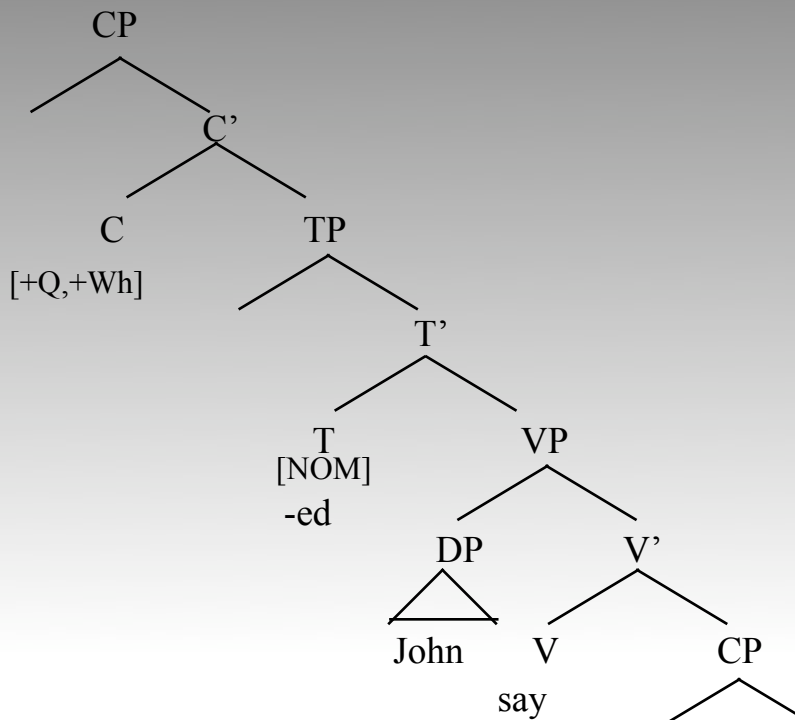
- 3) Wh-movement: John -ed say what was baked
- 4) DP-movement: John -ed say was baked what

We are going to tree this string (4): *the D-structure*

The D-structure



The D-structure



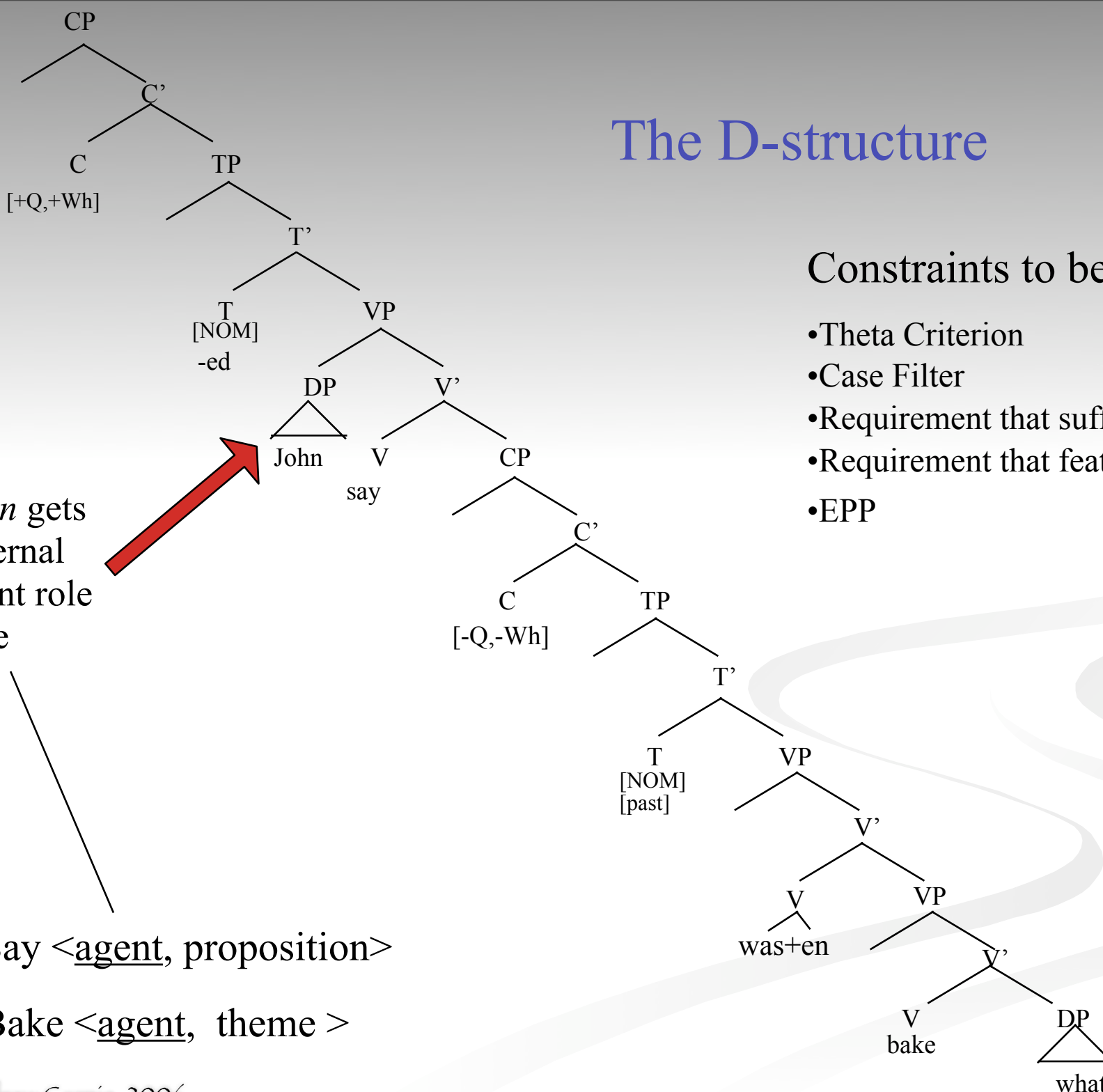
Constraints to be met

- Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP

• Say <agent, proposition>

• Bake <agent, theme >

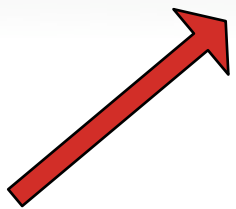
The D-structure



Constraints to be met

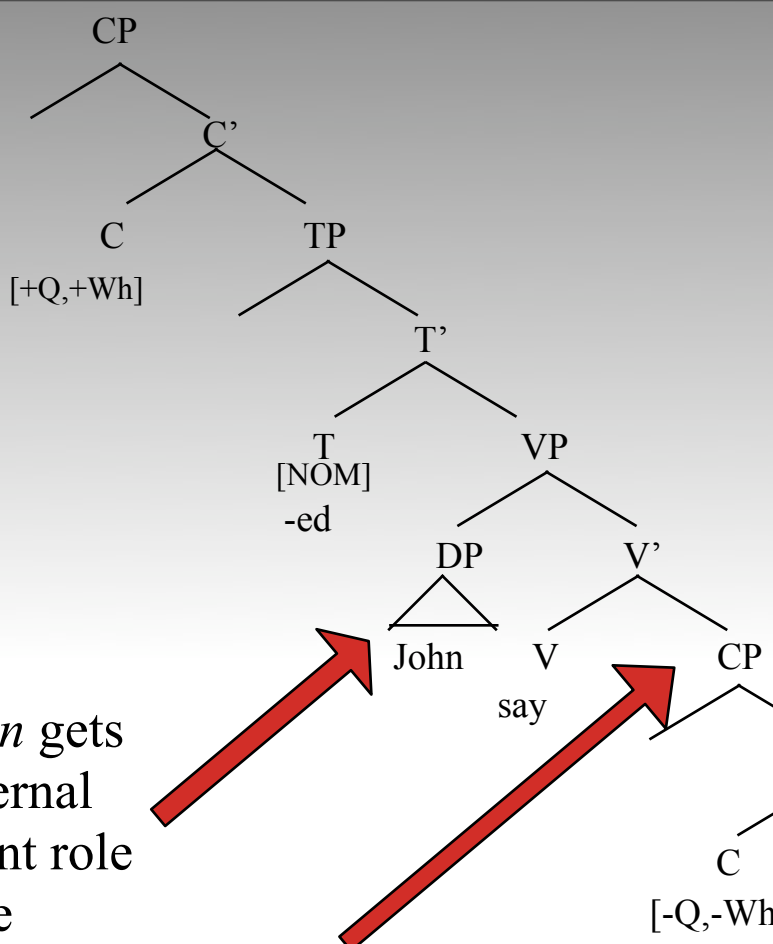
- Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP

John gets external agent role here



- Say <agent, proposition>
- Bake <agent, theme >

The D-structure



Constraints to be met

- Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP

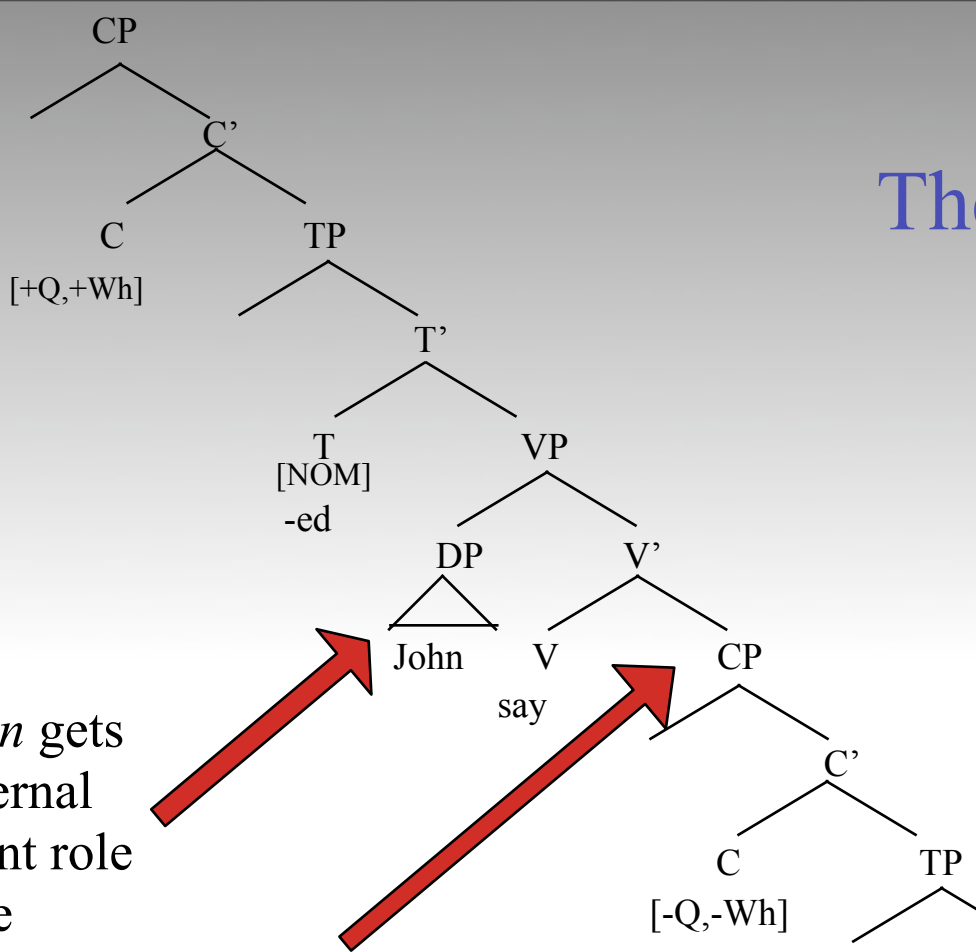
John gets external agent role here

CP gets internal proposition role here

• Say <agent, proposition>

• Bake <agent, theme >

The D-structure



Constraints to be met

- Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP

John gets external agent role here

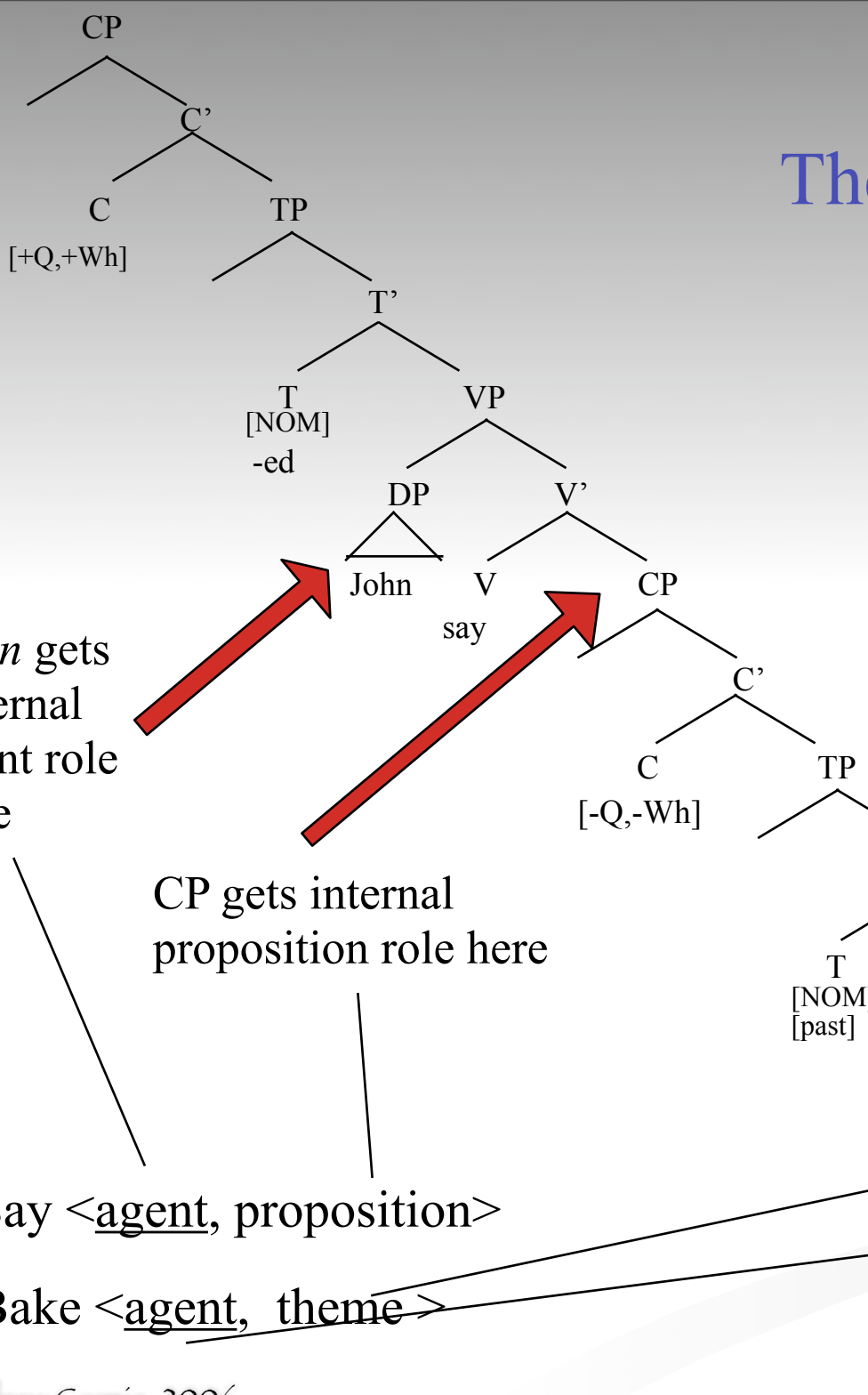
CP gets internal proposition role here

what gets internal theme theta role here

• Say < agent, proposition >

• Bake < agent, theme >

The D-structure



Constraints to be met

- Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP

John gets external agent role here

CP gets internal proposition role here

what gets internal theme theta role here

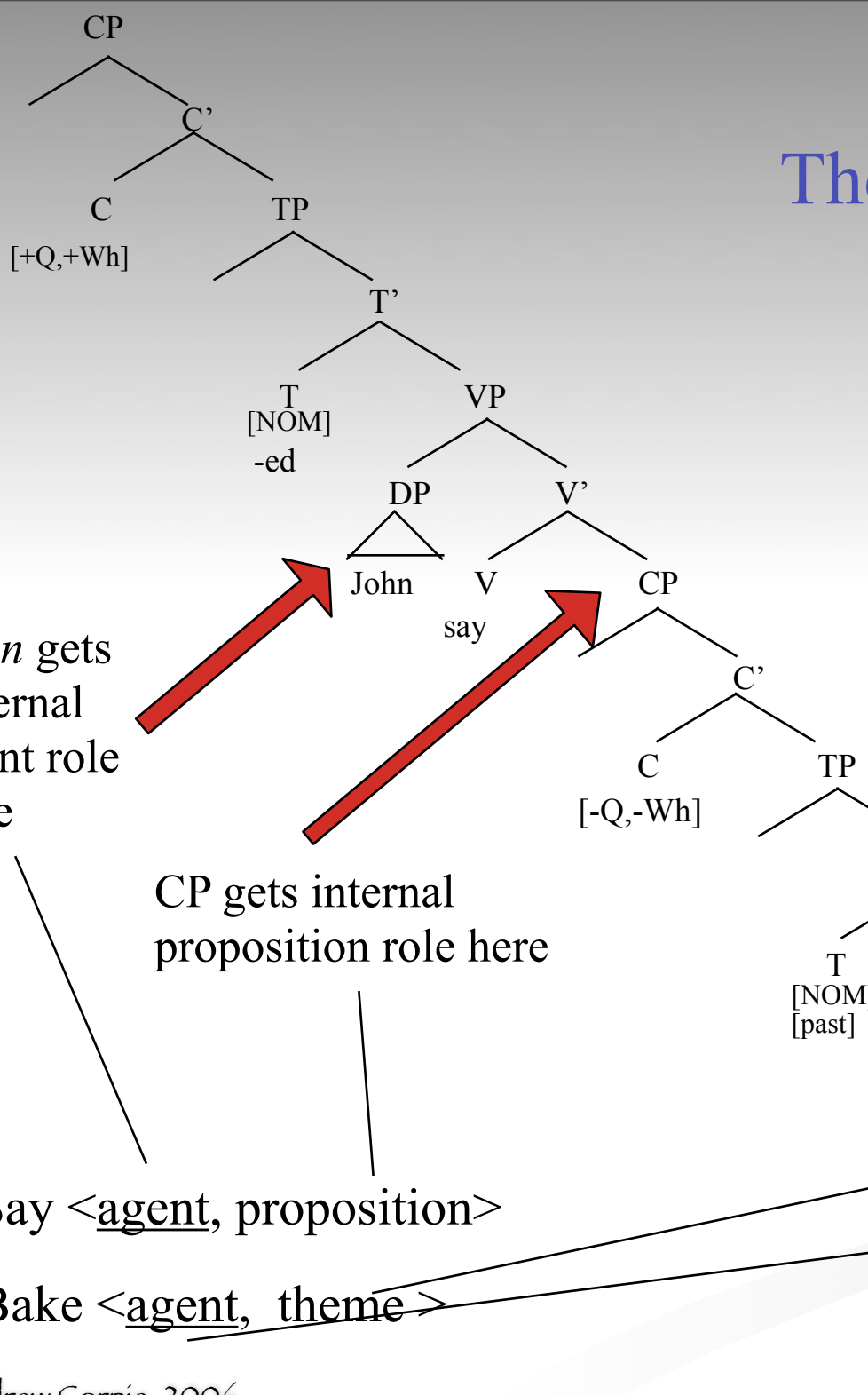
• Say <agent, proposition>

• Bake <agent, theme>

The D-structure

Constraints to be met

- ✓ • Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP



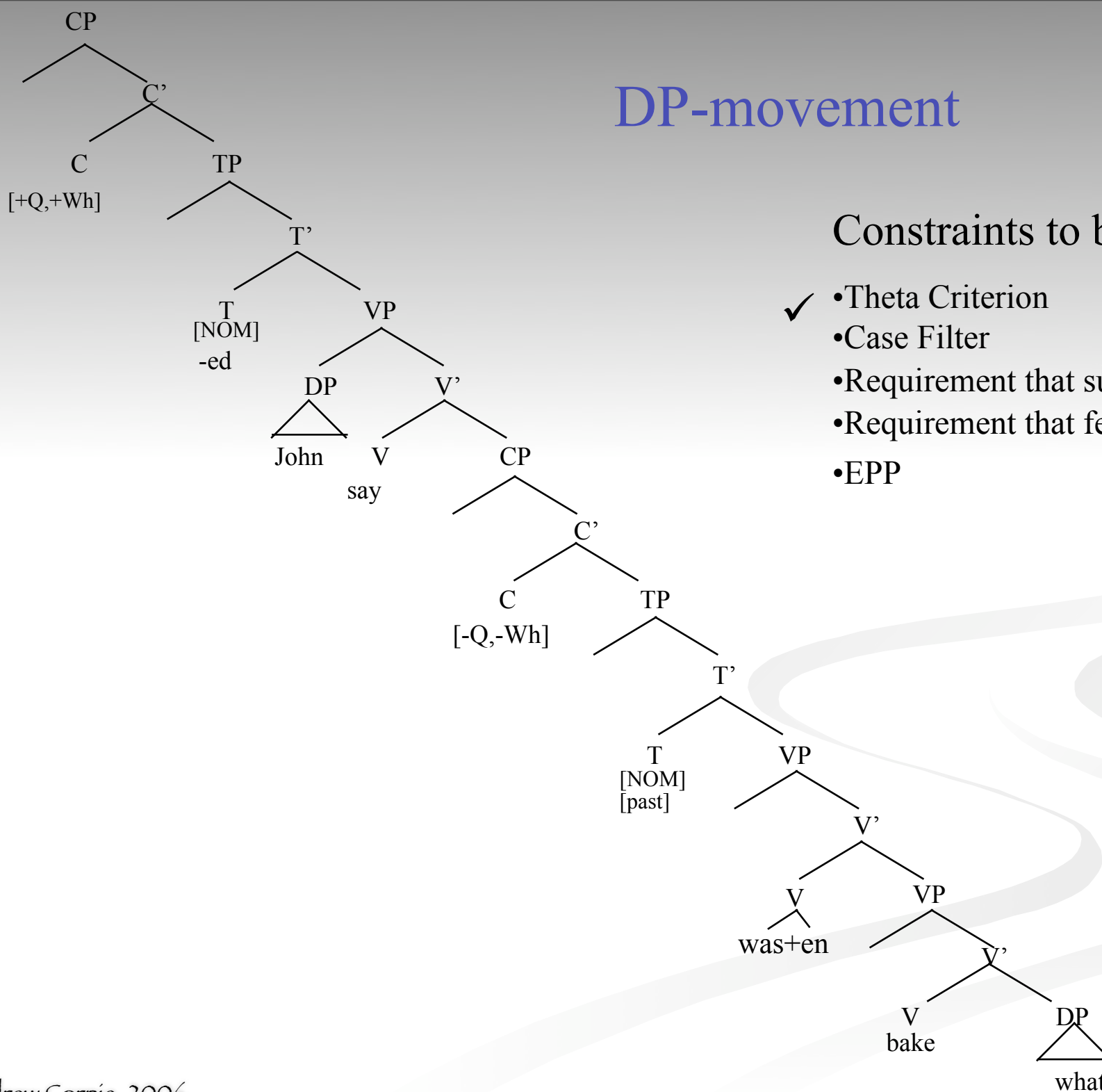
John gets external agent role here

CP gets internal proposition role here

what gets internal theme theta role here

- Say <agent, proposition>
- Bake <agent, theme>

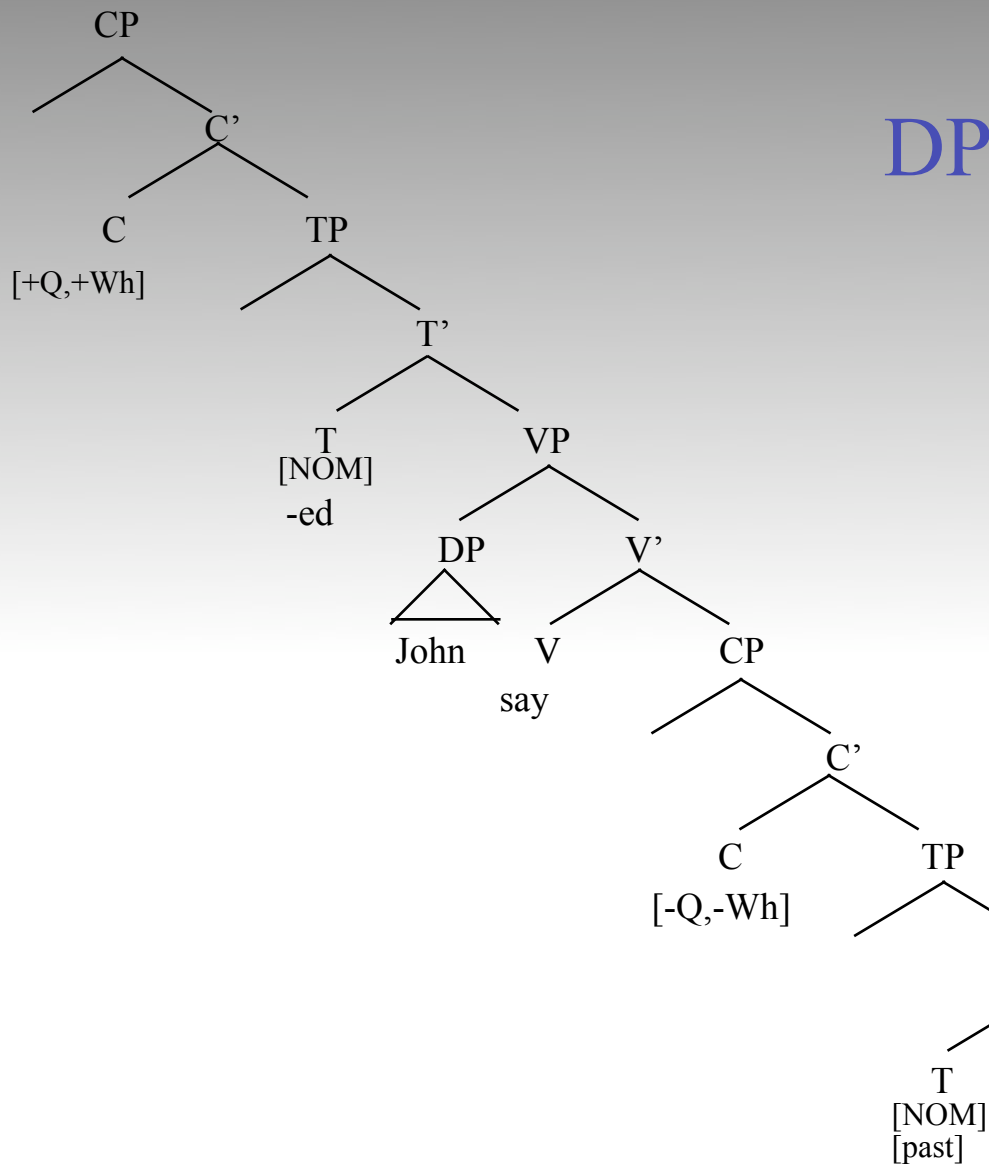
DP-movement



Constraints to be met

- ✓ •Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP

DP-movement



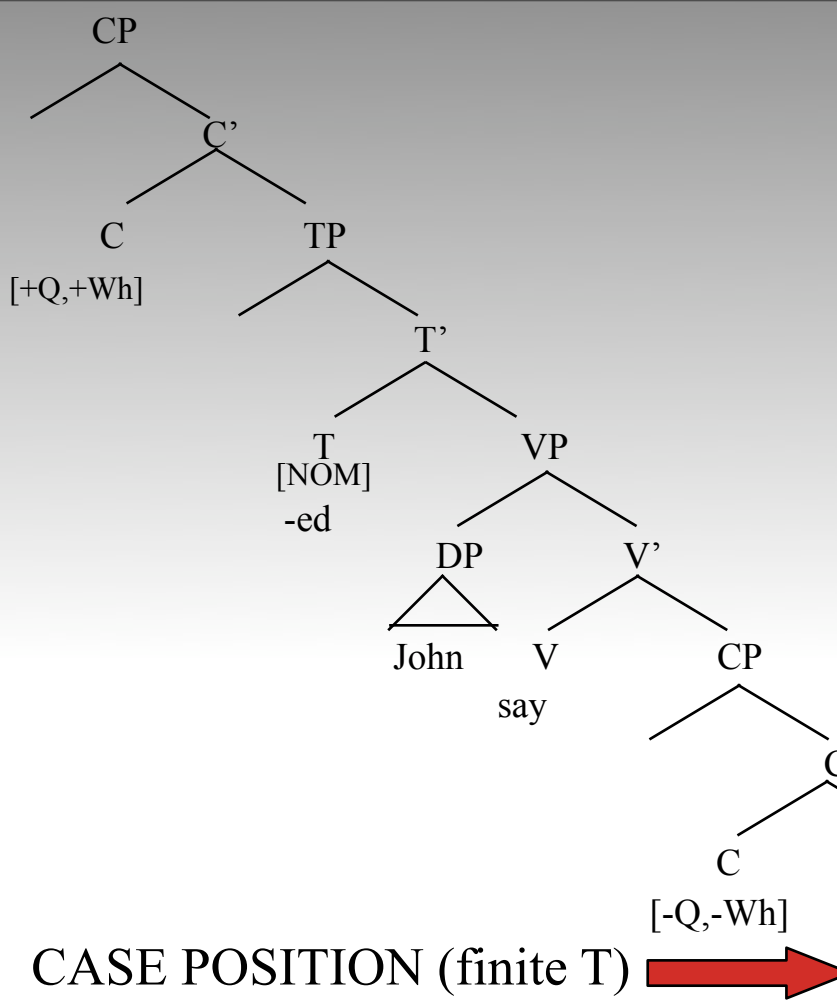
Constraints to be met

- ✓ •Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP

NOT A CASE POSITION
(passive verb)



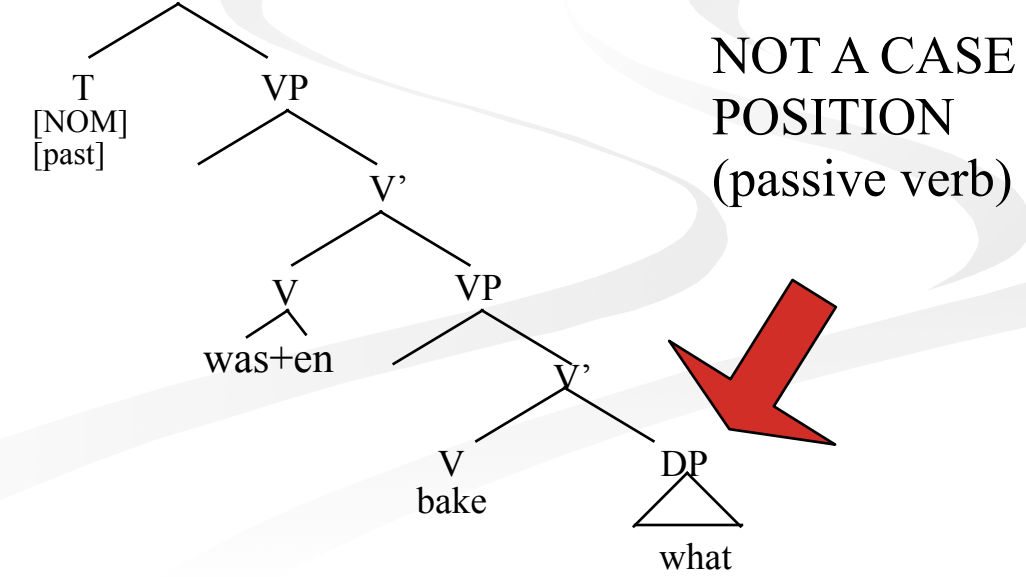
DP-movement



Constraints to be met

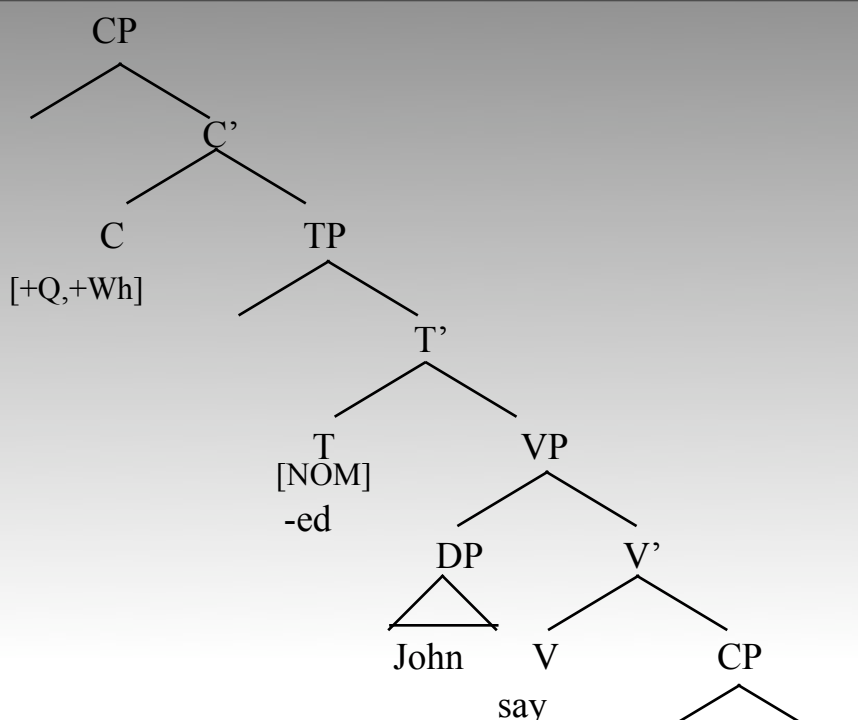
- ✓ •Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP

CASE POSITION (finite T) 



NOT A CASE POSITION (passive verb) 

DP-movement

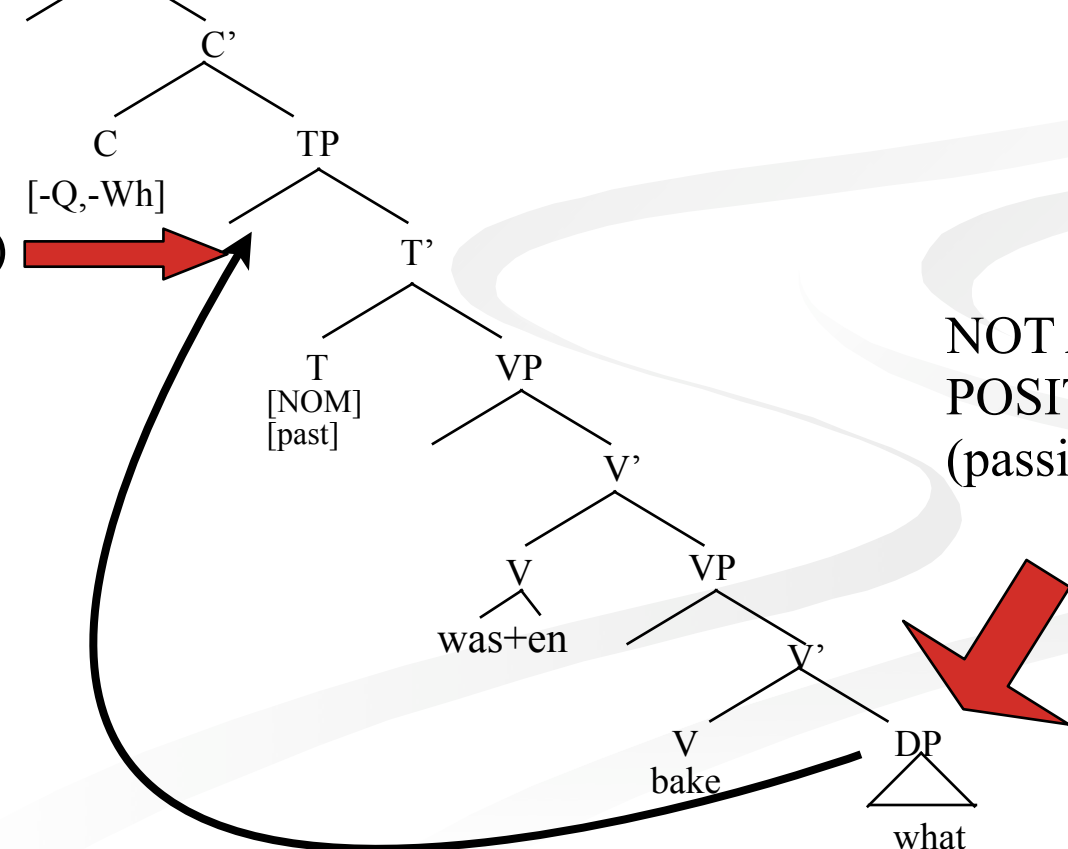


Constraints to be met

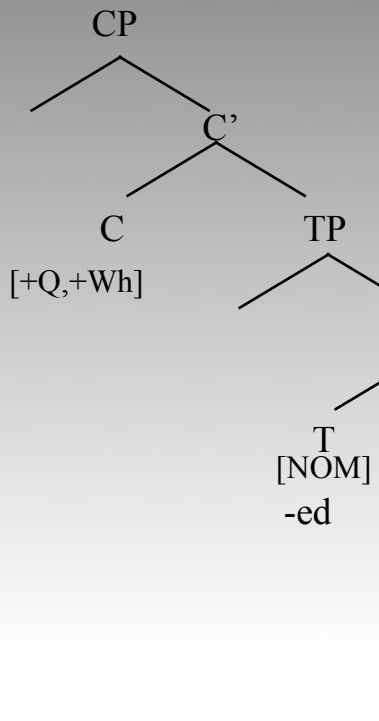
- ✓ •Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP

CASE POSITION (finite T)

NOT A CASE POSITION (passive verb)



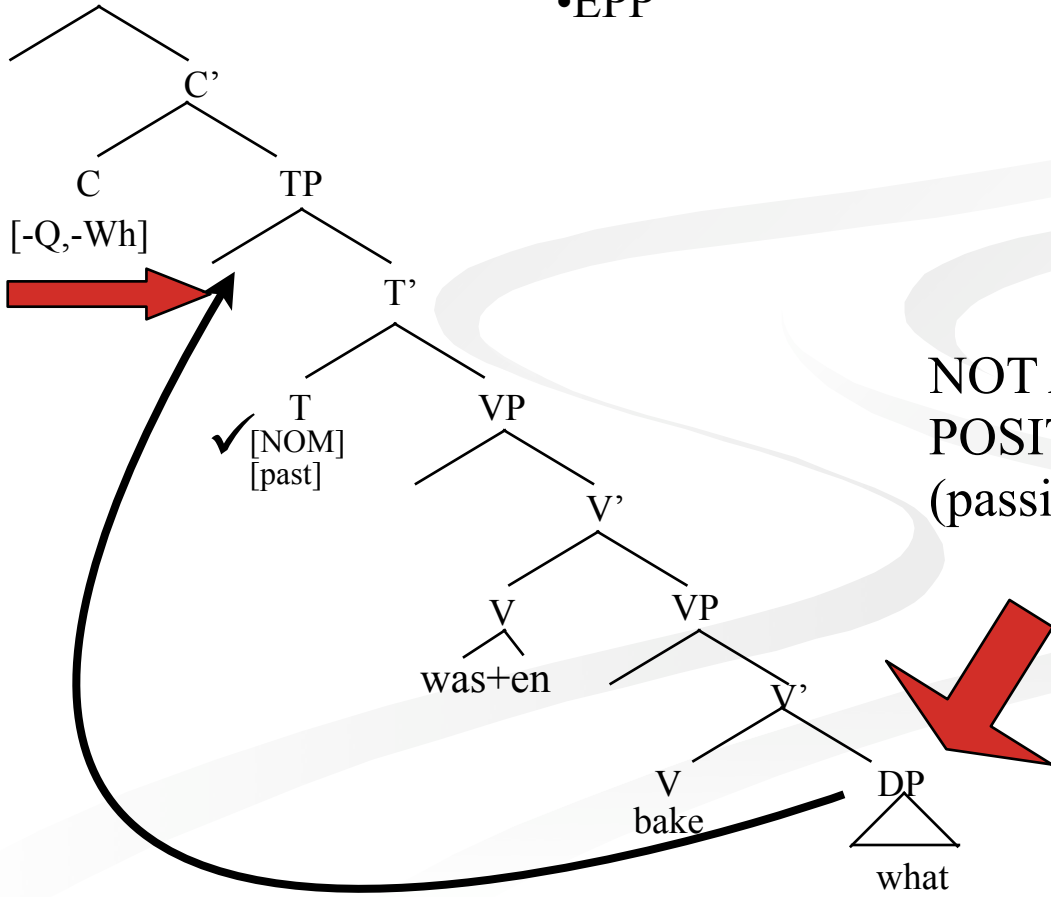
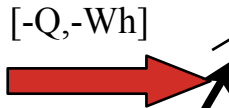
DP-movement



Constraints to be met

- ✓ •Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP

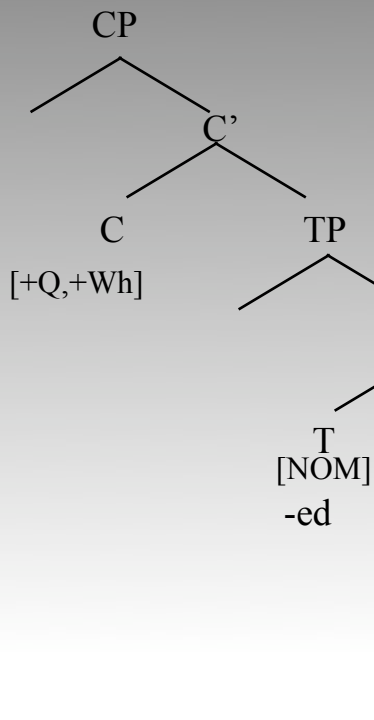
CASE POSITION (finite T)



NOT A CASE POSITION (passive verb)



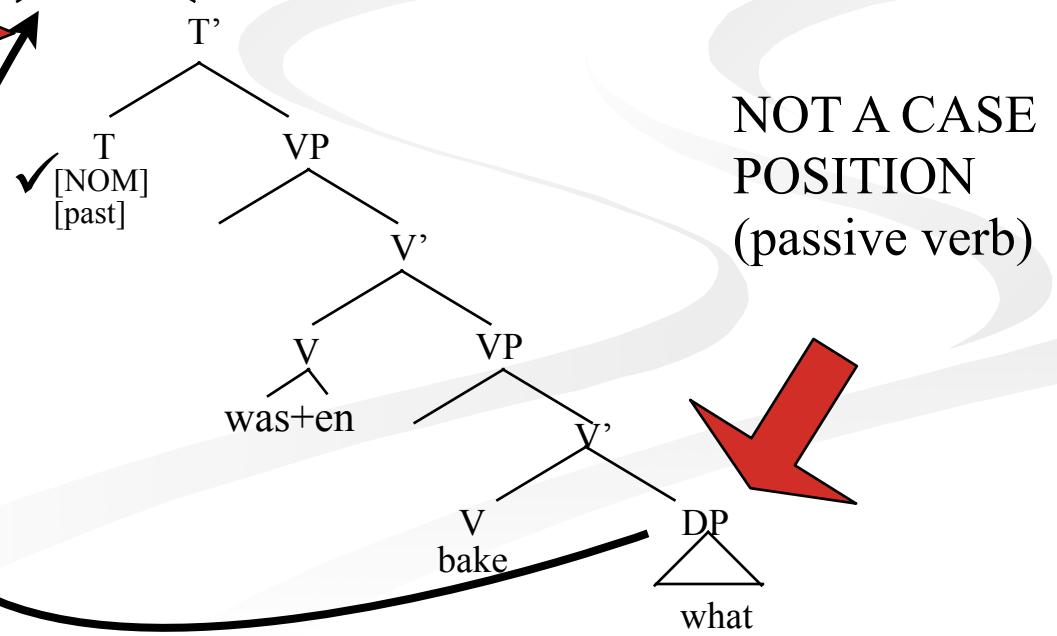
DP-movement



Constraints to be met

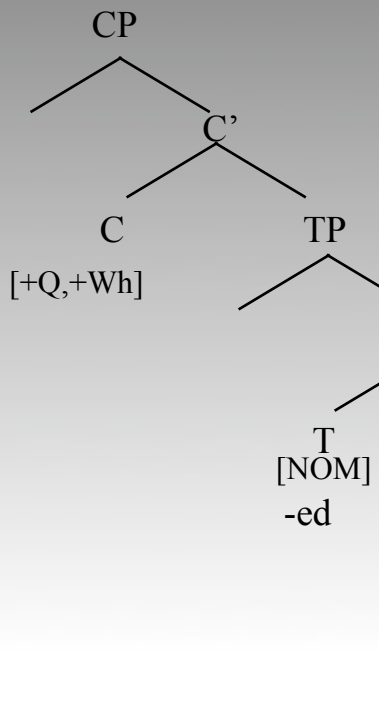
- ✓ •Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP

CASE POSITION (finite T)



Can we just skip this step? No -- would be a case filter violation?

DP-movement

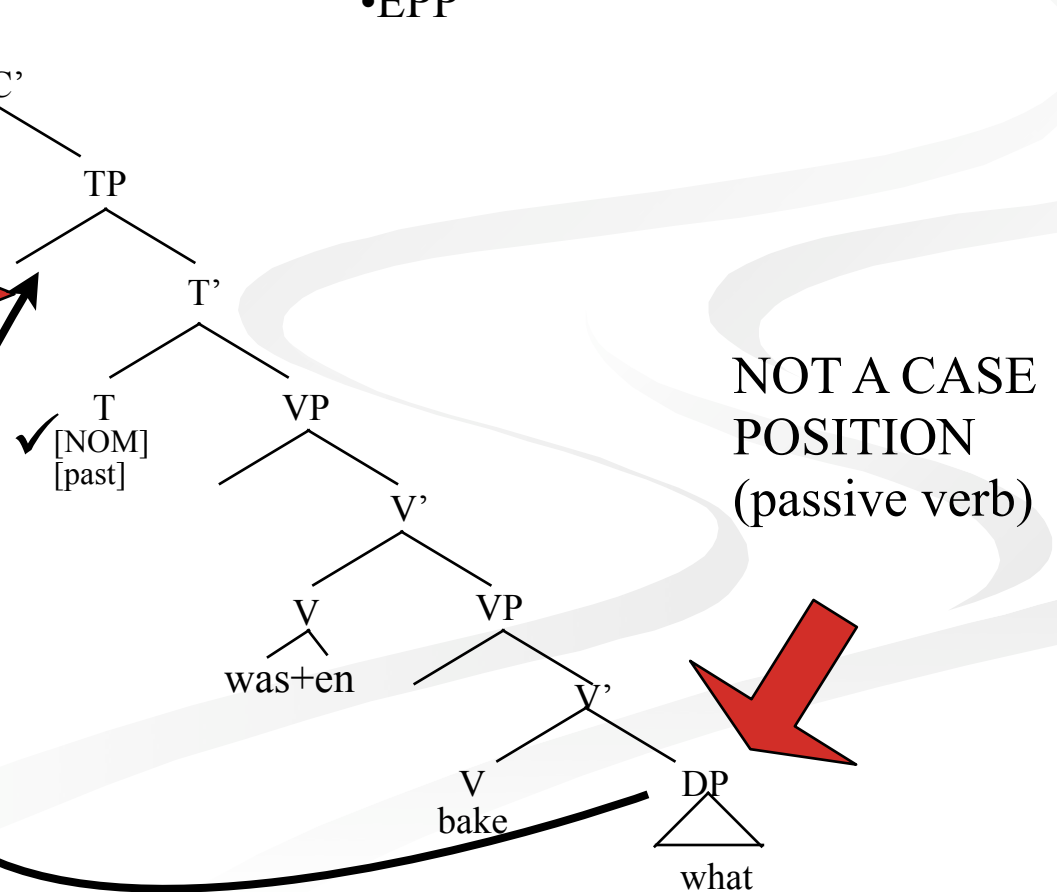
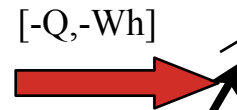


NOT A CASE POSITION (spec,VP)

Constraints to be met

- ✓ •Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP

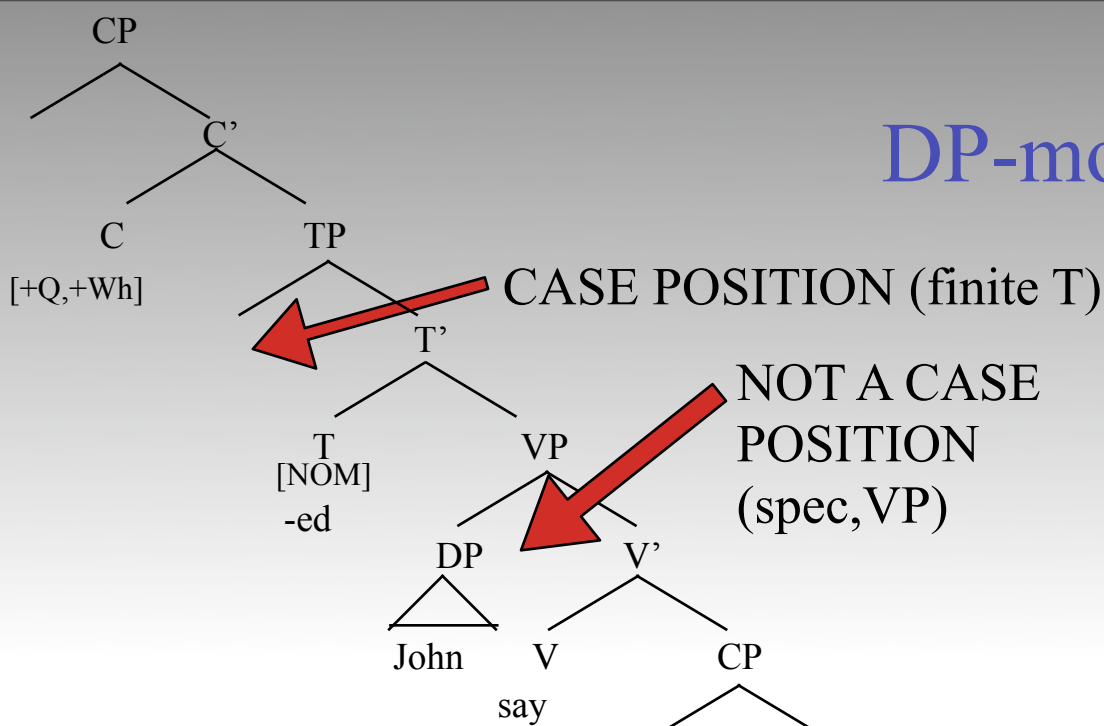
CASE POSITION (finite T)



NOT A CASE POSITION (passive verb)

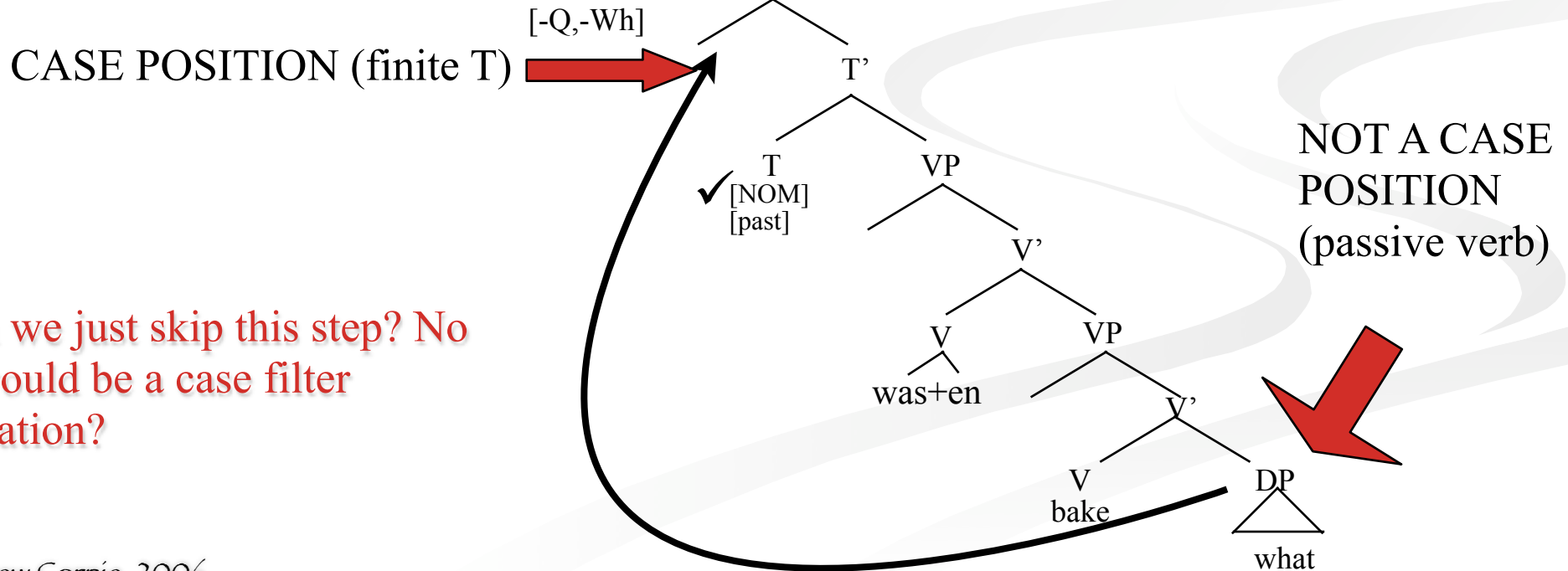
Can we just skip this step? No -- would be a case filter violation?

DP-movement



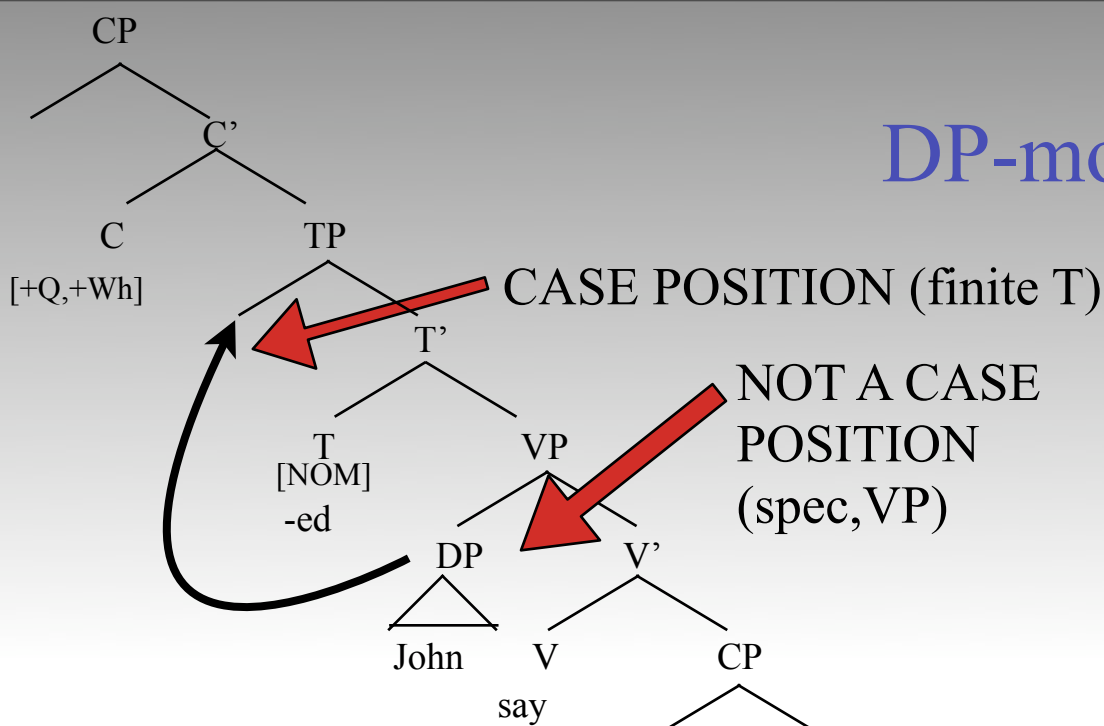
Constraints to be met

- ✓ •Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP



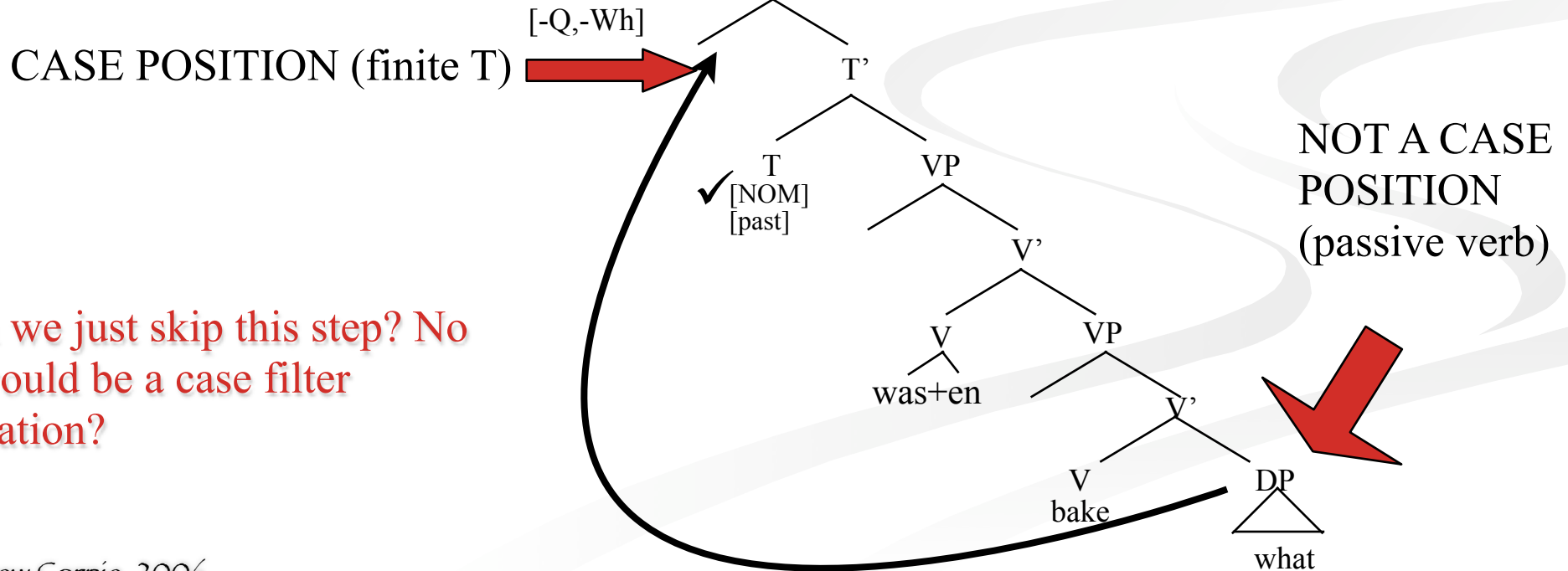
Can we just skip this step? No -- would be a case filter violation?

DP-movement



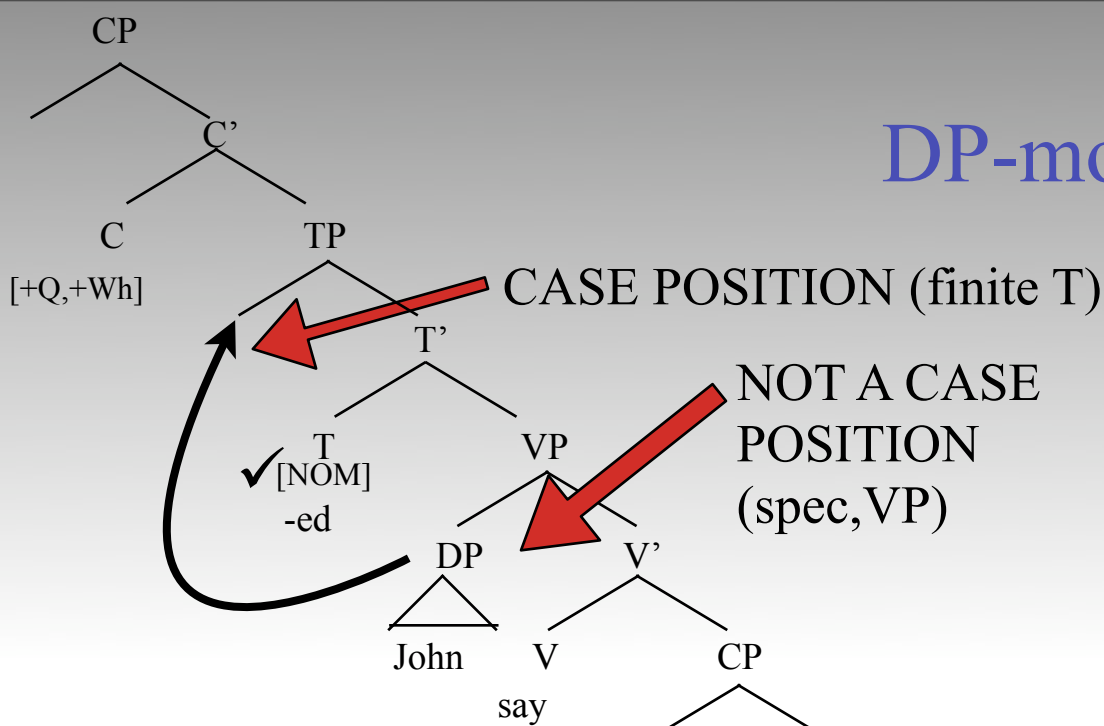
Constraints to be met

- ✓ •Theta Criterion
- Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP



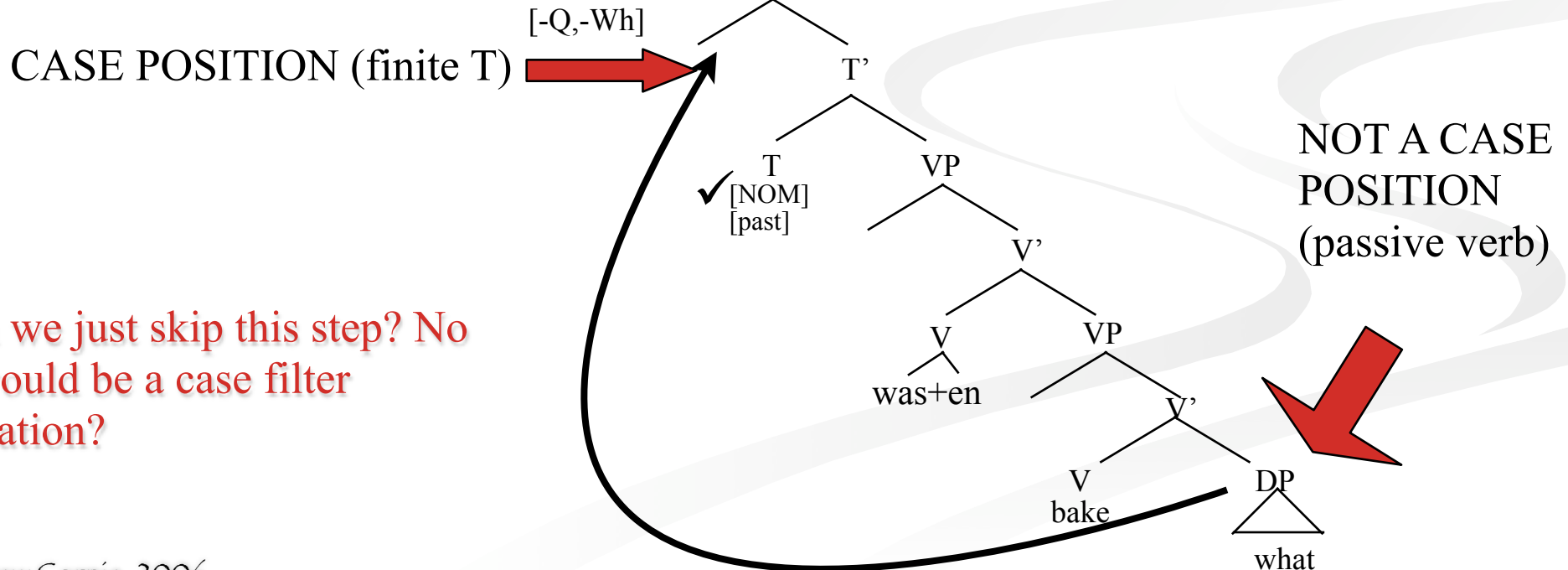
Can we just skip this step? No -- would be a case filter violation?

DP-movement



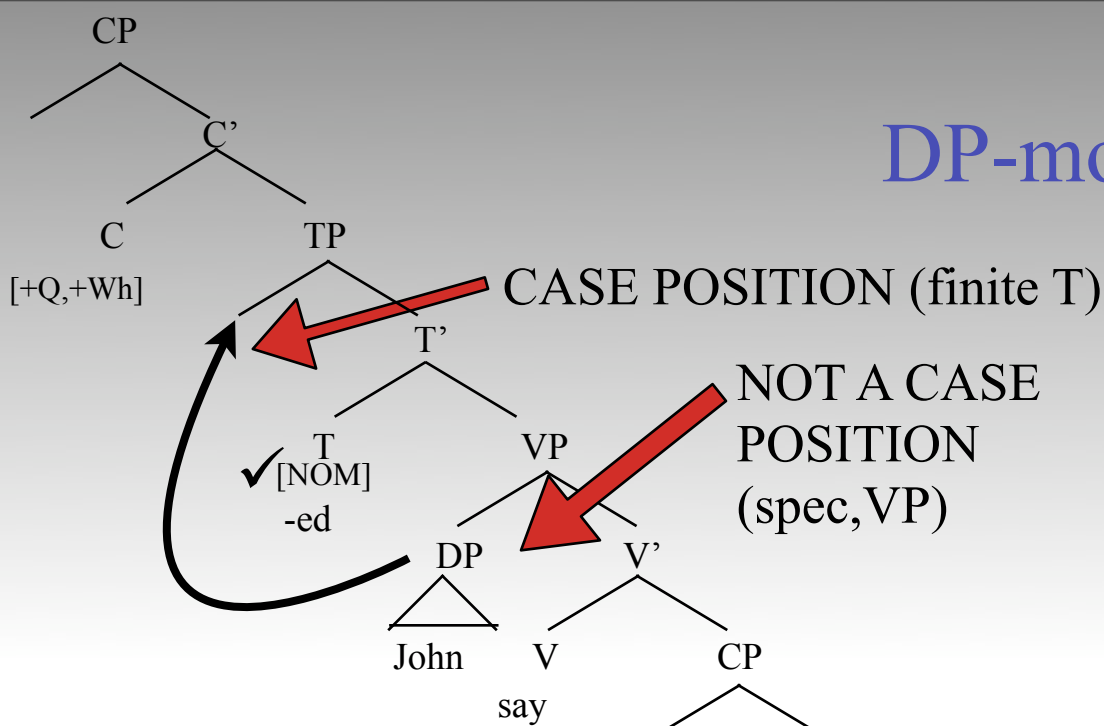
Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP



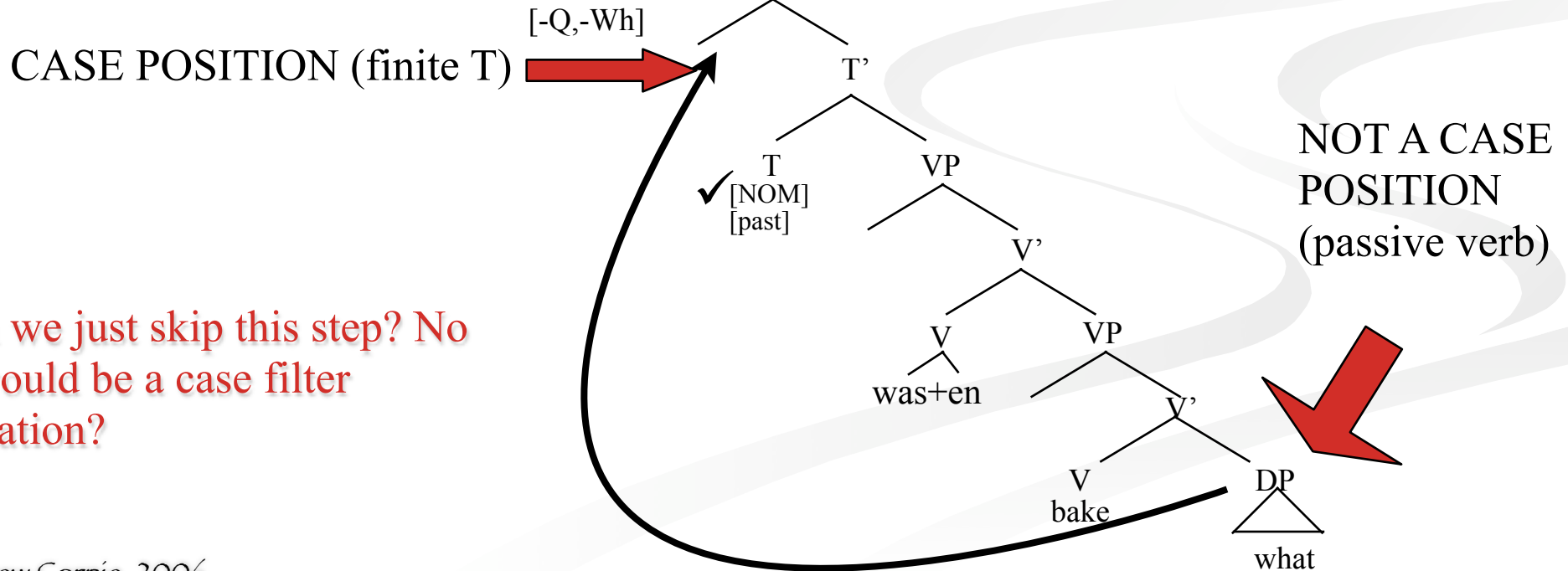
Can we just skip this step? No -- would be a case filter violation?

DP-movement



Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP



Can we just skip this step? No -- would be a case filter violation?

An aside: the order of things



An aside: the order of things

- START in your theta position



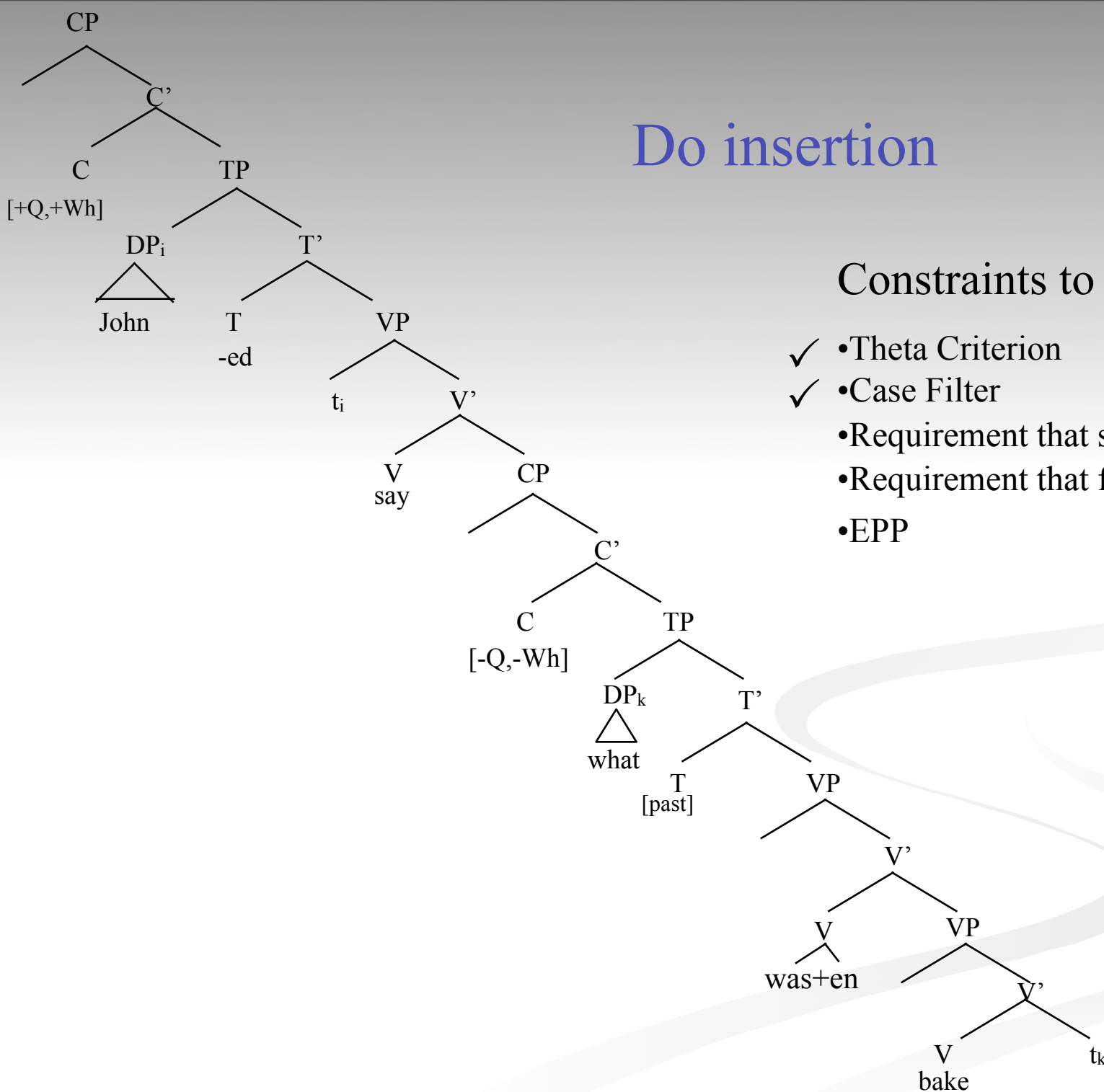
An aside: the order of things

- START in your theta position
- If necessary, DP-move to get case. (*DP movement always ends in a case position*)

An aside: the order of things

- START in your theta position
- If necessary, DP-move to get case. (*DP movement always ends in a case position*)
- Then if necessary, Wh-move to the specifier of CP to check [+wh]. (*Wh-movement always starts in a case position*)

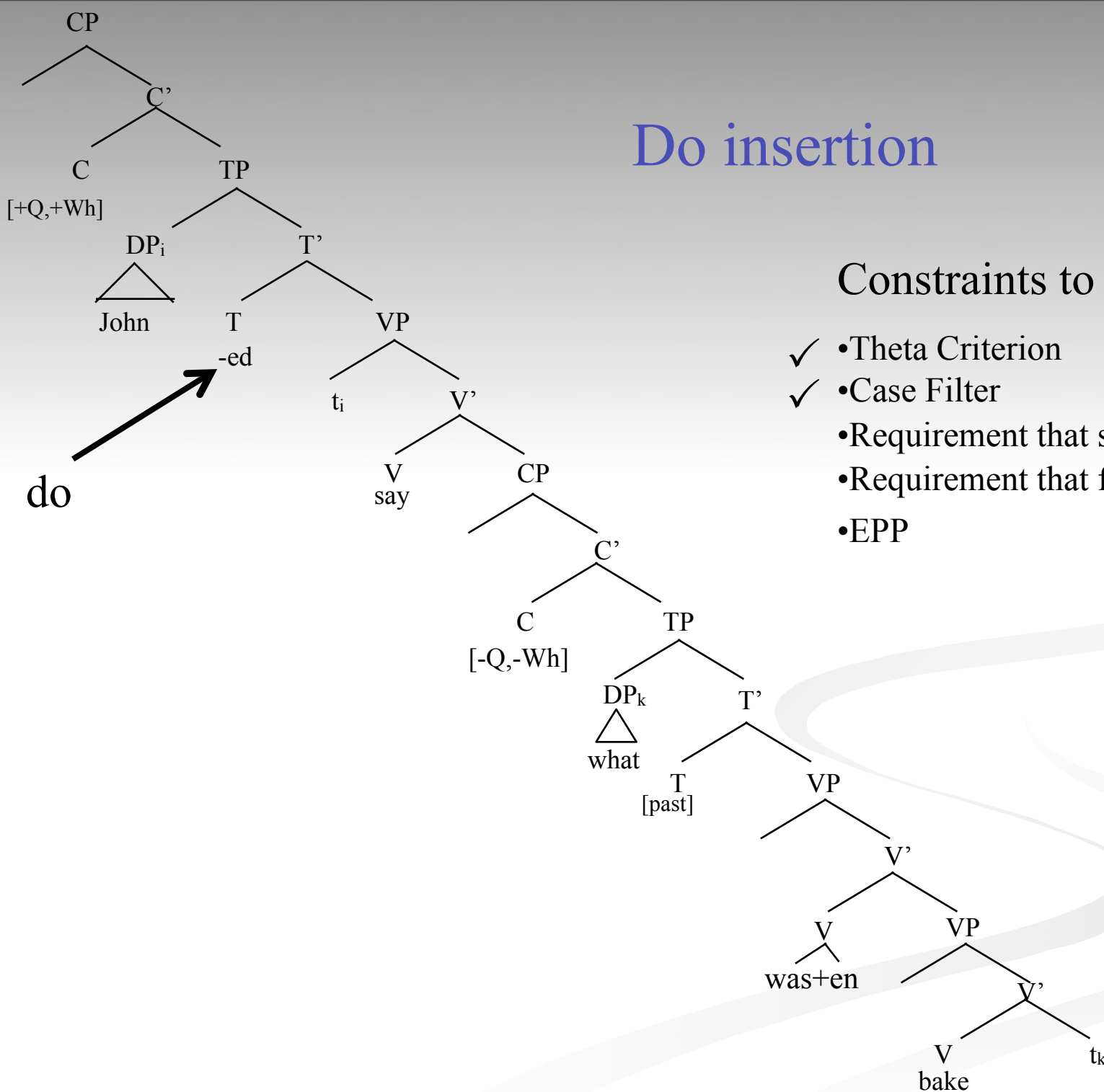
Do insertion



Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP

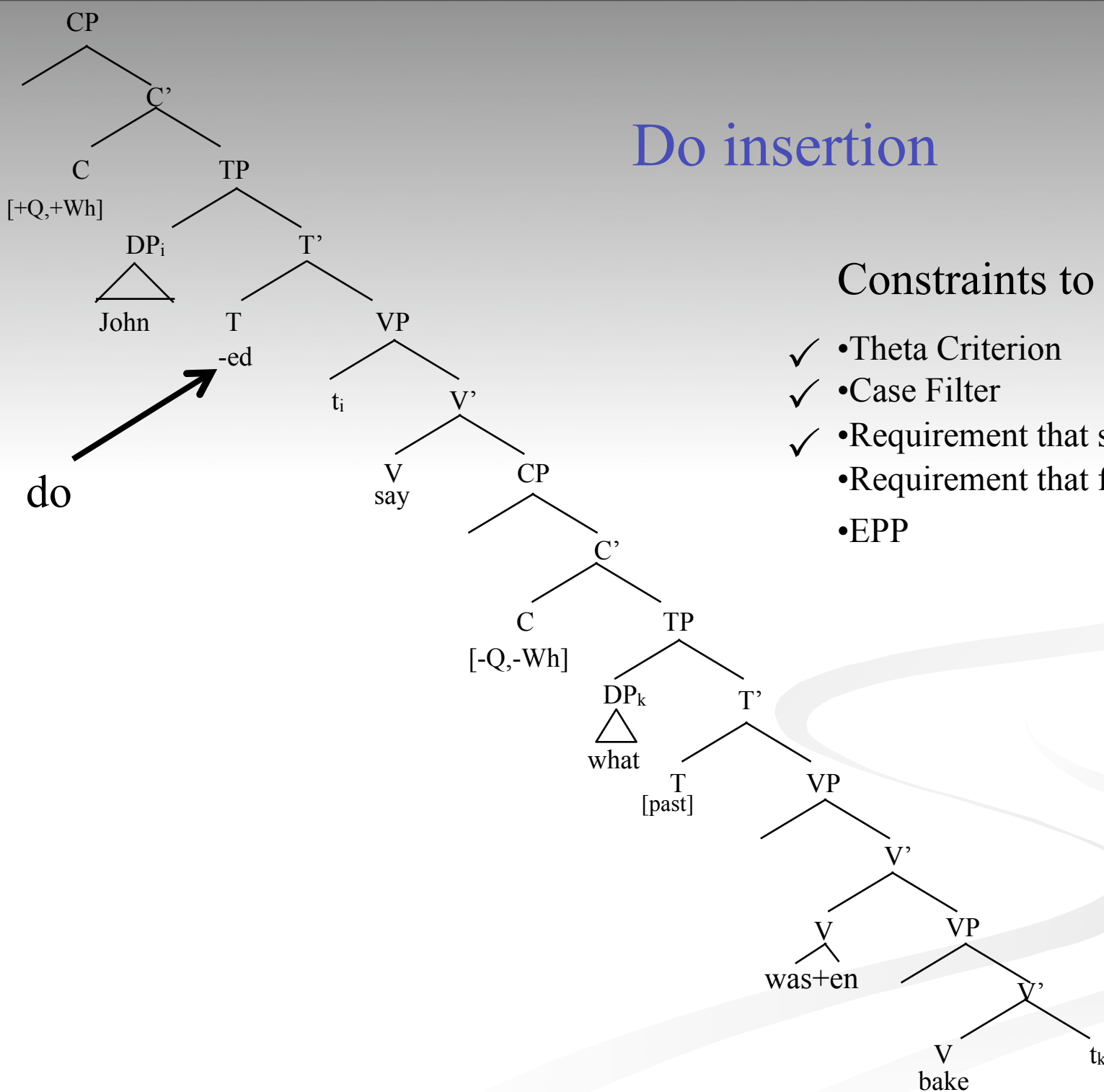
Do insertion



Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- Requirement that suffixes are attached
- Requirement that features are checked
- EPP

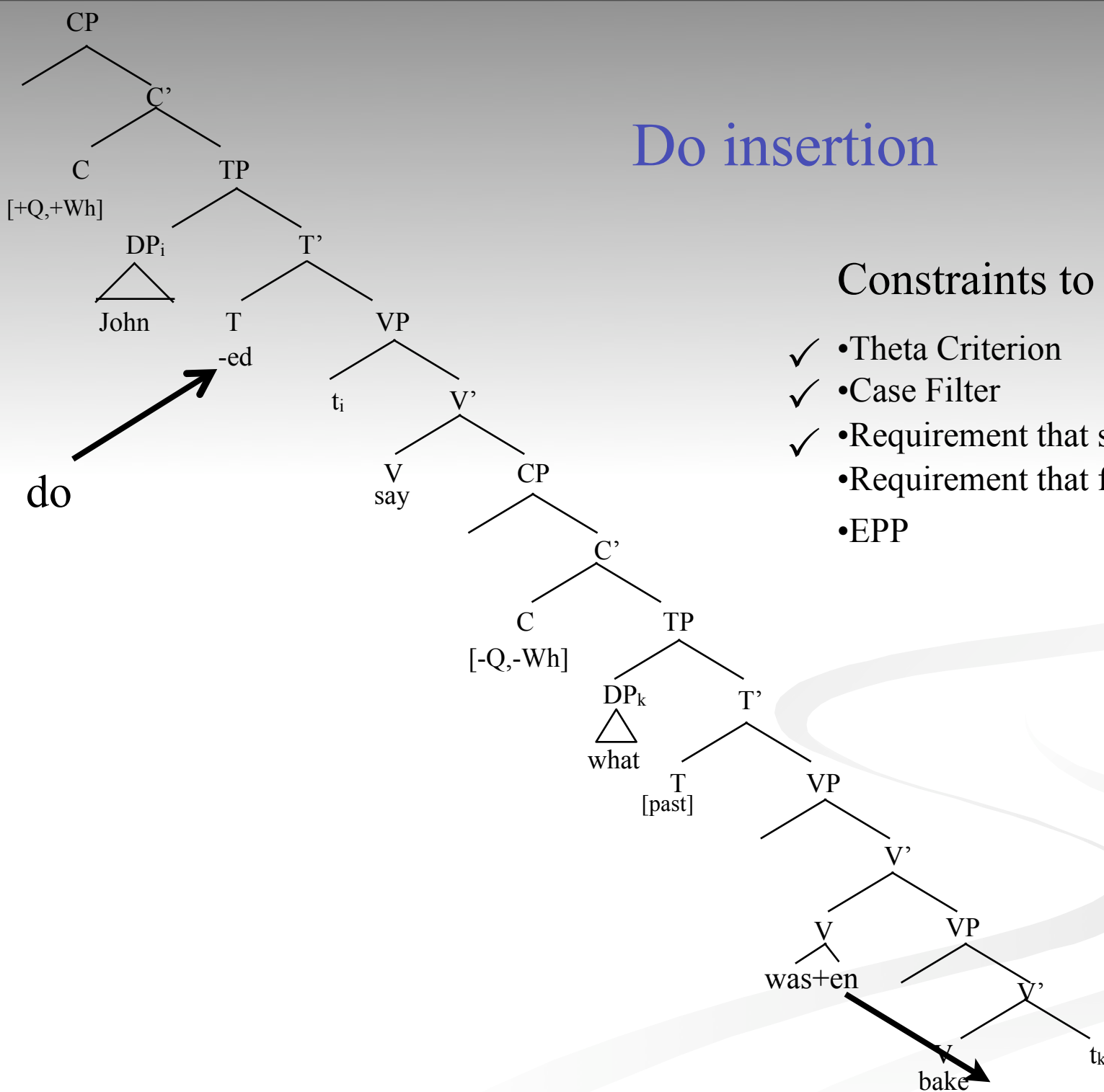
Do insertion



Constraints to be met

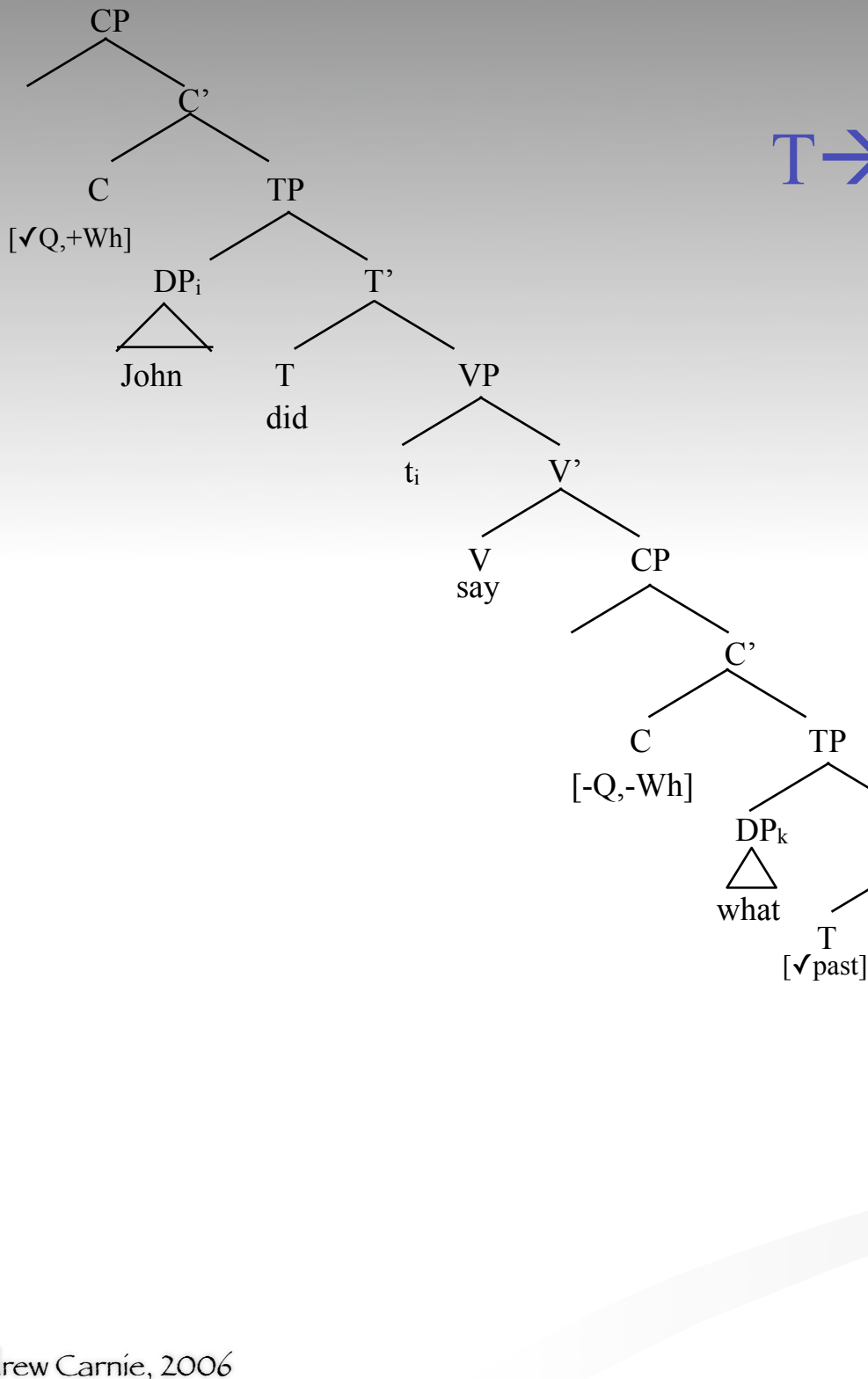
- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- Requirement that features are checked
- EPP

Do insertion



- ### Constraints to be met
- ✓ •Theta Criterion
 - ✓ •Case Filter
 - ✓ •Requirement that suffixes are attached
 - Requirement that features are checked
 - EPP

T → C movement

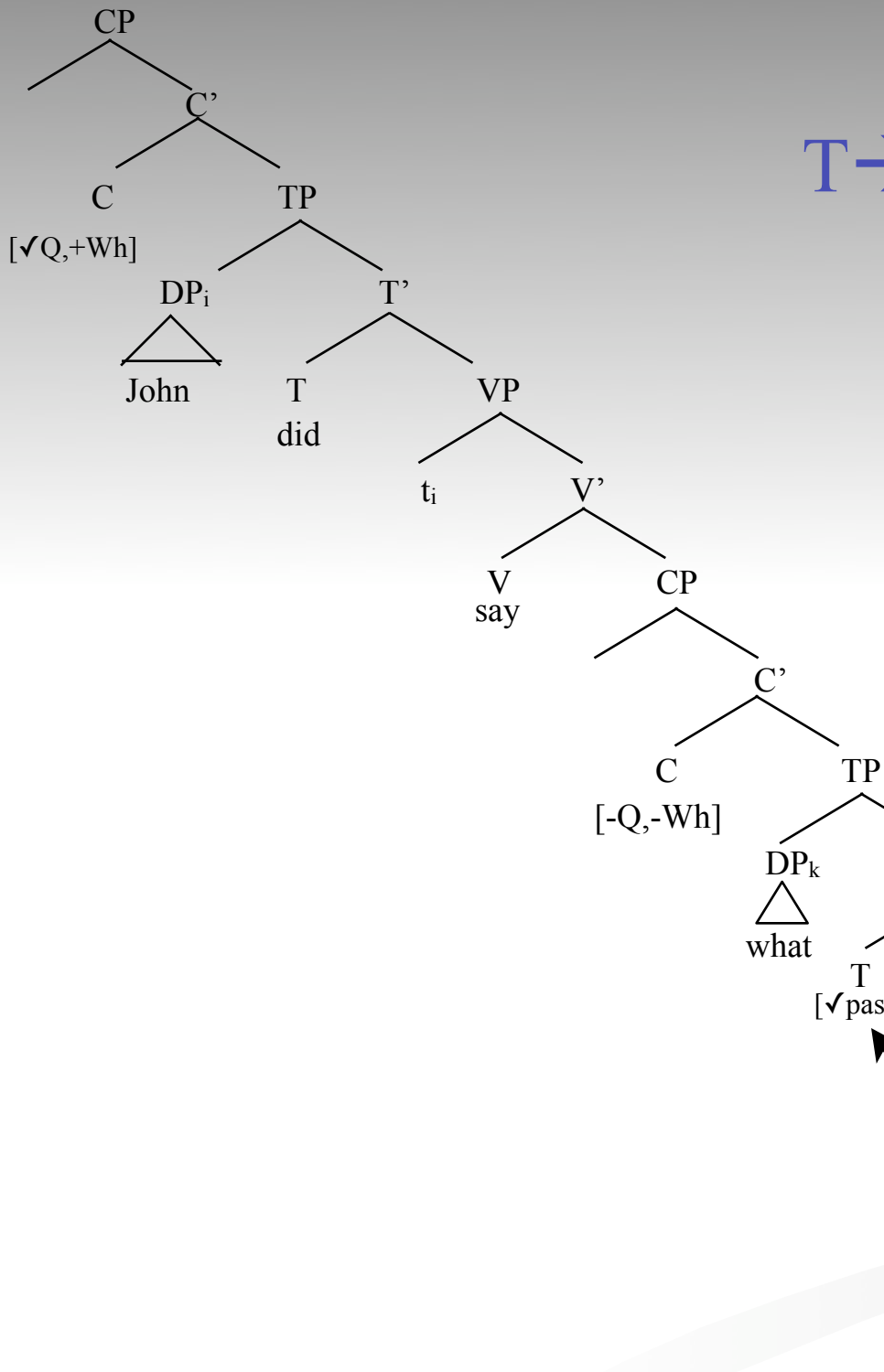


Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- Requirement that features are checked
 - [present]
 - [+Q]
 - [+Wh]
- EPP

⇒baked

T → C movement

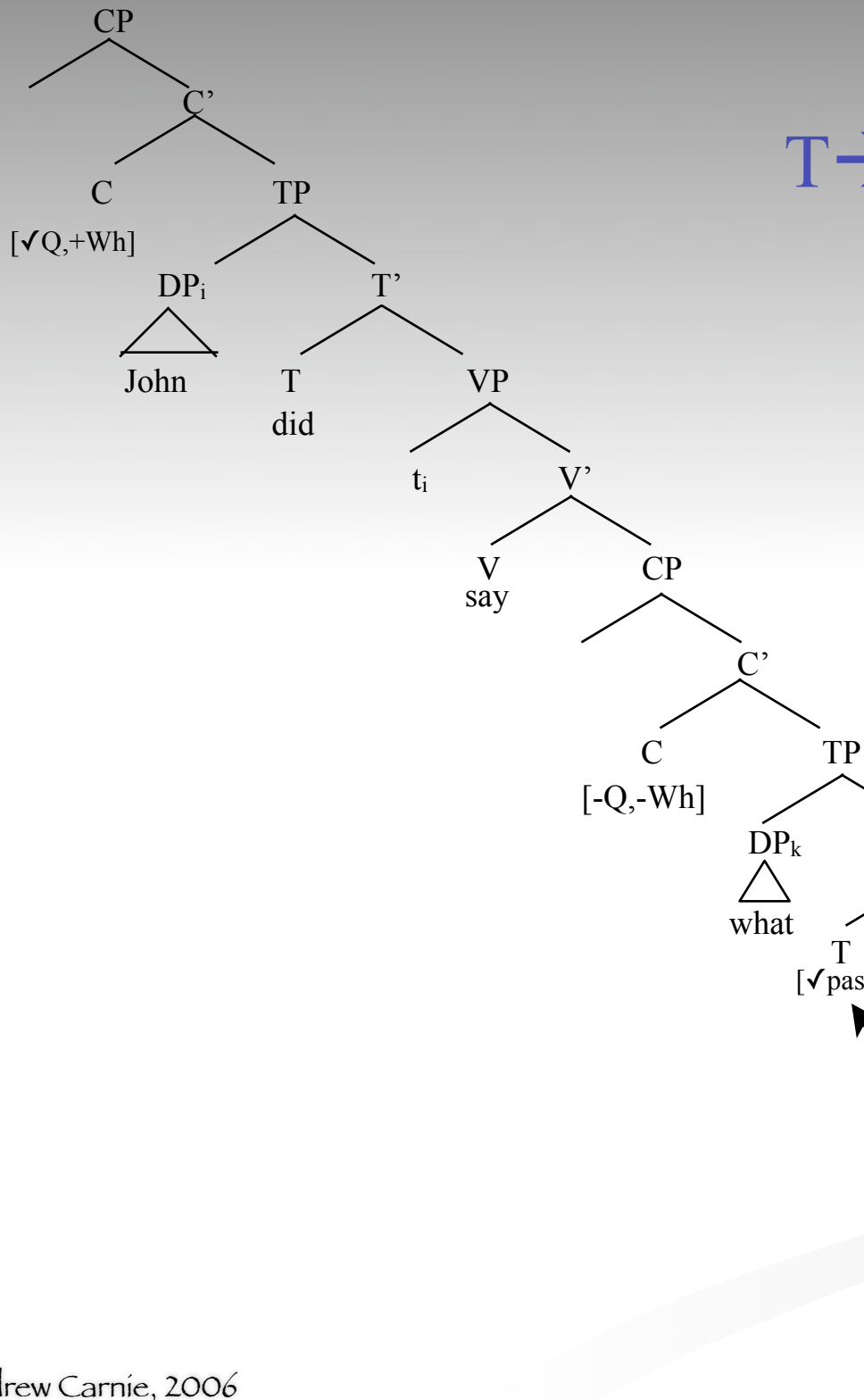


Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- Requirement that features are checked
 - [present]
 - [+Q]
 - [+Wh]
- EPP

⇒baked

T → C movement

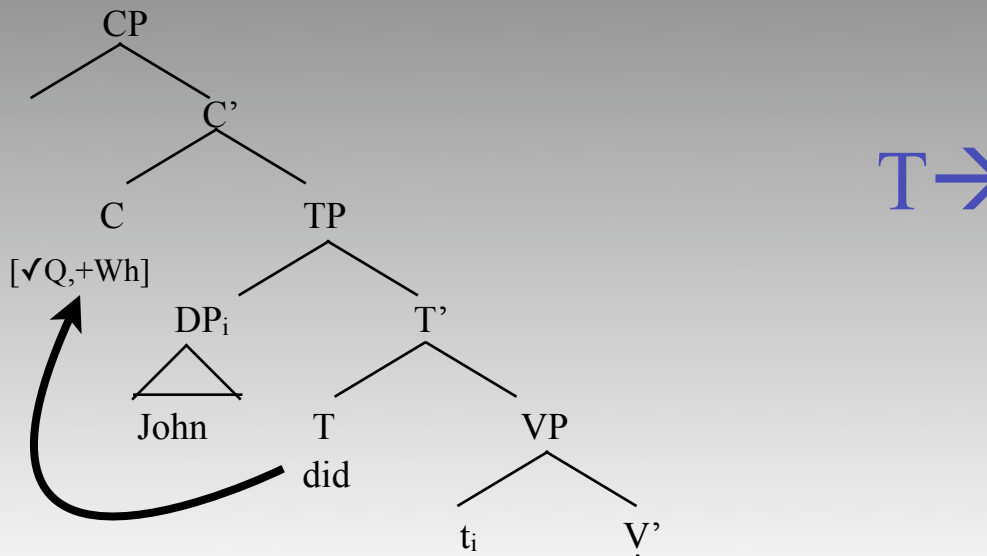


Constraints to be met

- ✓ • Theta Criterion
- ✓ • Case Filter
- ✓ • Requirement that suffixes are attached
- Requirement that features are checked
 - ✓ • [present]
 - [+Q]
 - [+Wh]
- EPP

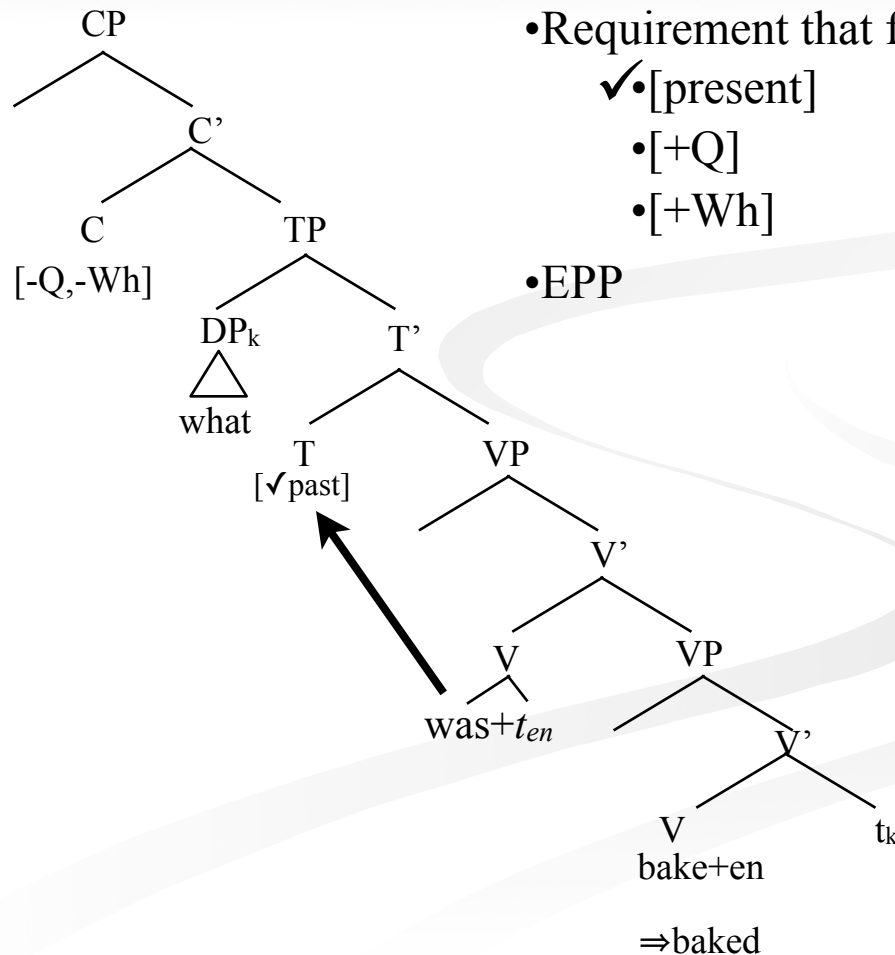
⇒baked

T → C movement

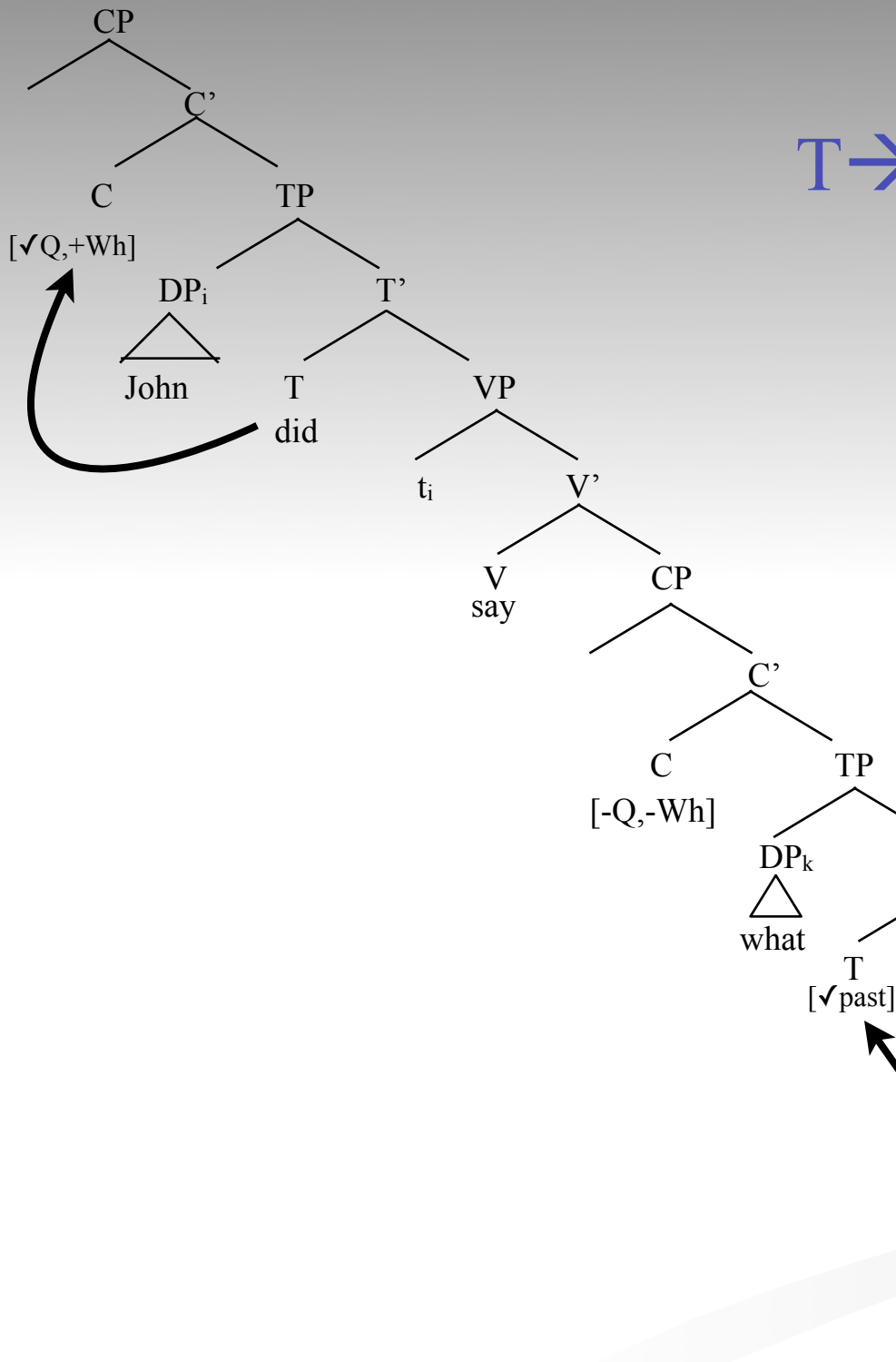


Constraints to be met

- ✓ • Theta Criterion
- ✓ • Case Filter
- ✓ • Requirement that suffixes are attached
- Requirement that features are checked
 - ✓ • [present]
 - [+Q]
 - [+Wh]
- EPP



T → C movement

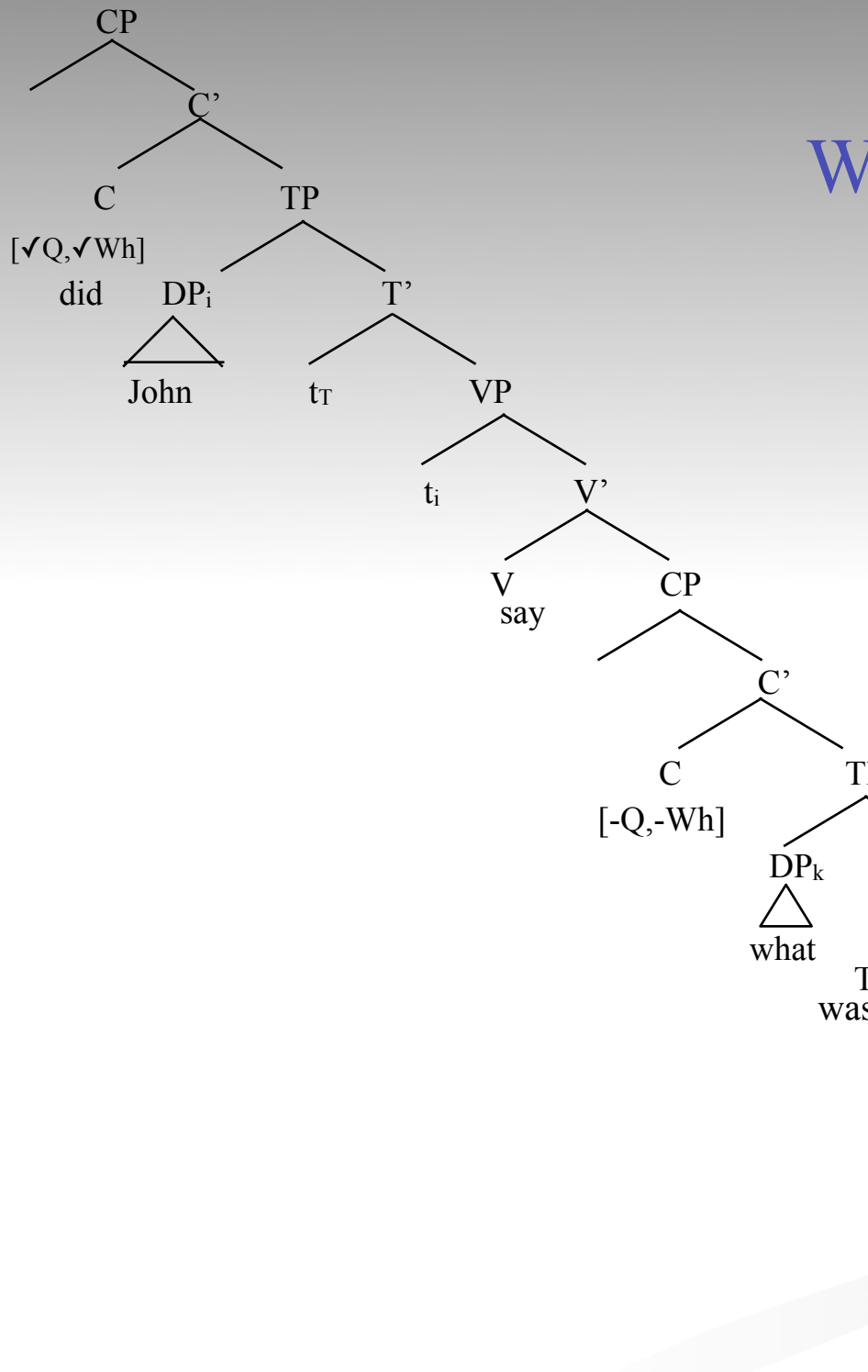


Constraints to be met

- ✓ • Theta Criterion
- ✓ • Case Filter
- ✓ • Requirement that suffixes are attached
- Requirement that features are checked
 - ✓ • [present]
 - ✓ • [+Q]
 - [+Wh]
- EPP

⇒ baked

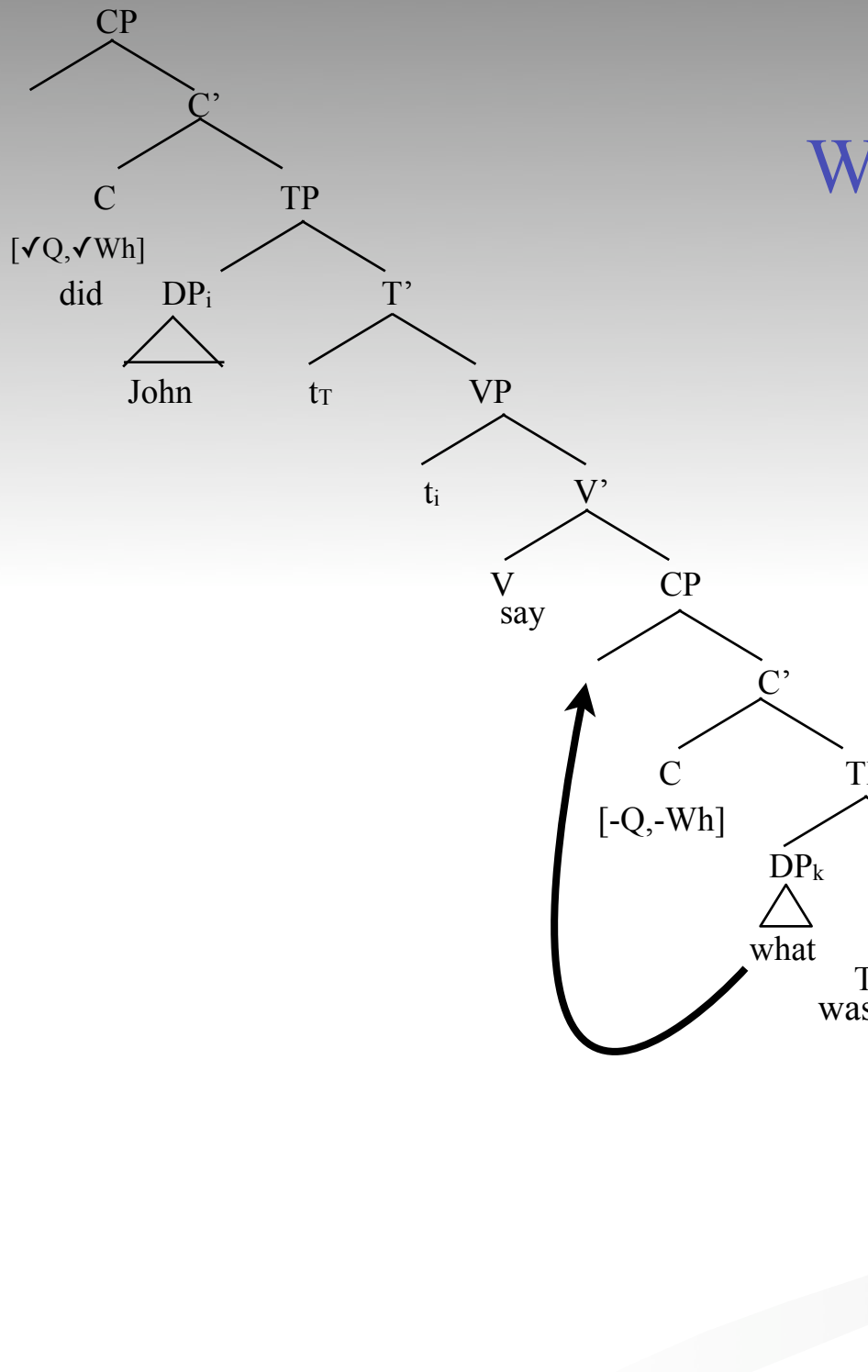
Wh-movement



Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- Requirement that features are checked
 - ✓•[past]
 - ✓•[+Q]
 - [+Wh]
- EPP

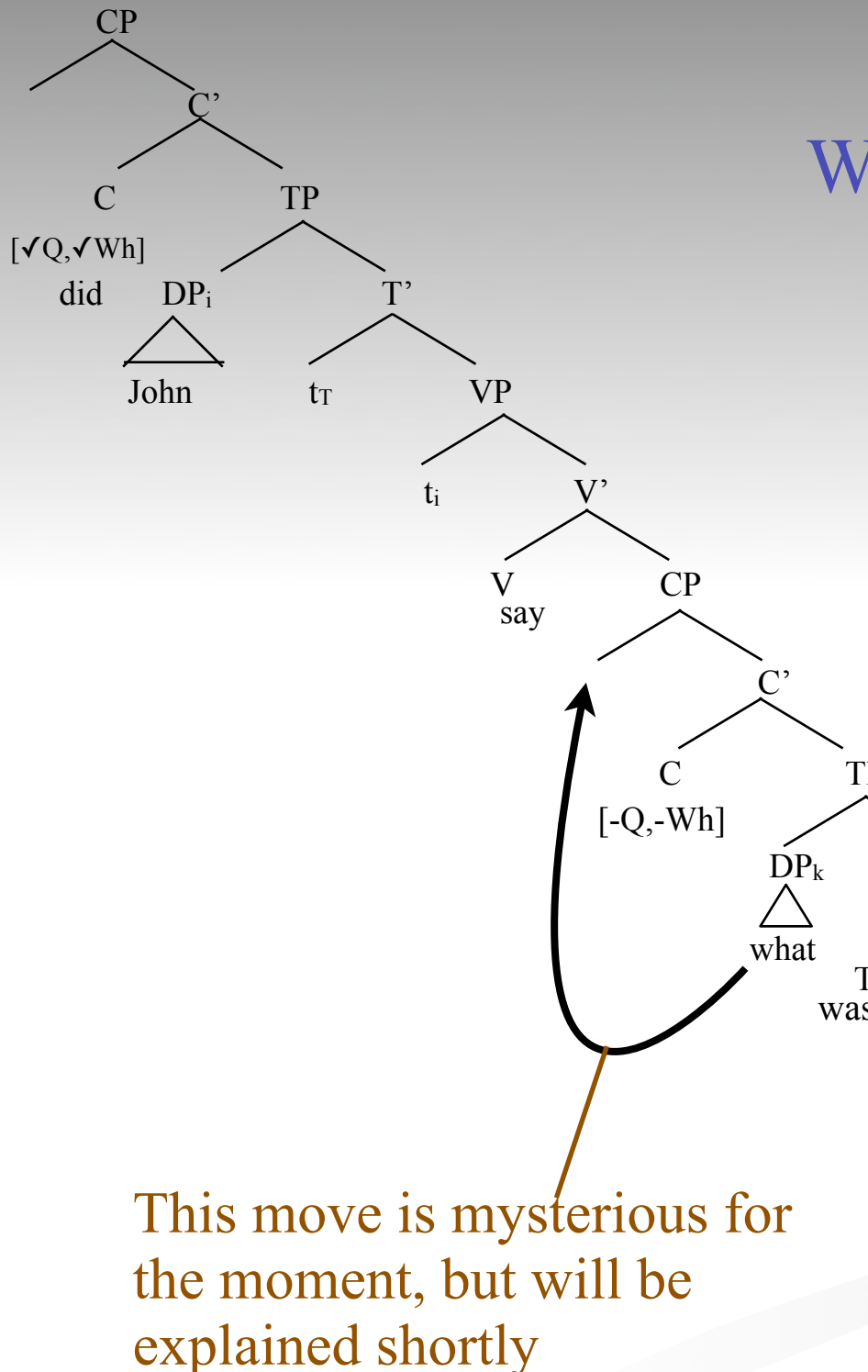
Wh-movement



Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- Requirement that features are checked
 - ✓•[past]
 - ✓•[+Q]
 - [+Wh]
- EPP

Wh-movement



Constraints to be met

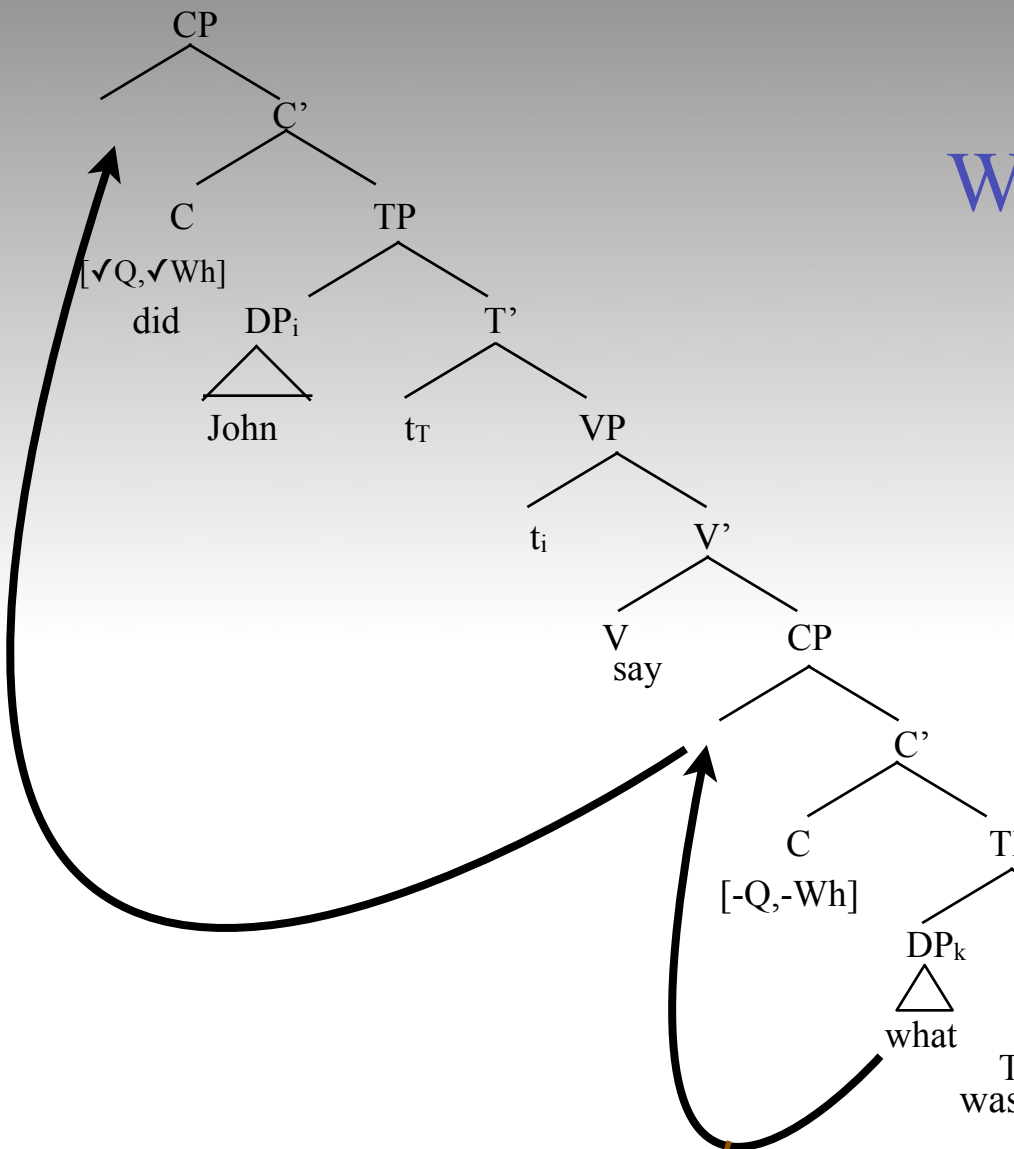
- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- Requirement that features are checked
 - ✓•[past]
 - ✓•[+Q]
 - [+Wh]
- EPP

This move is mysterious for the moment, but will be explained shortly

Wh-movement

Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- Requirement that features are checked
 - ✓•[past]
 - ✓•[+Q]
 - [+Wh]
- EPP

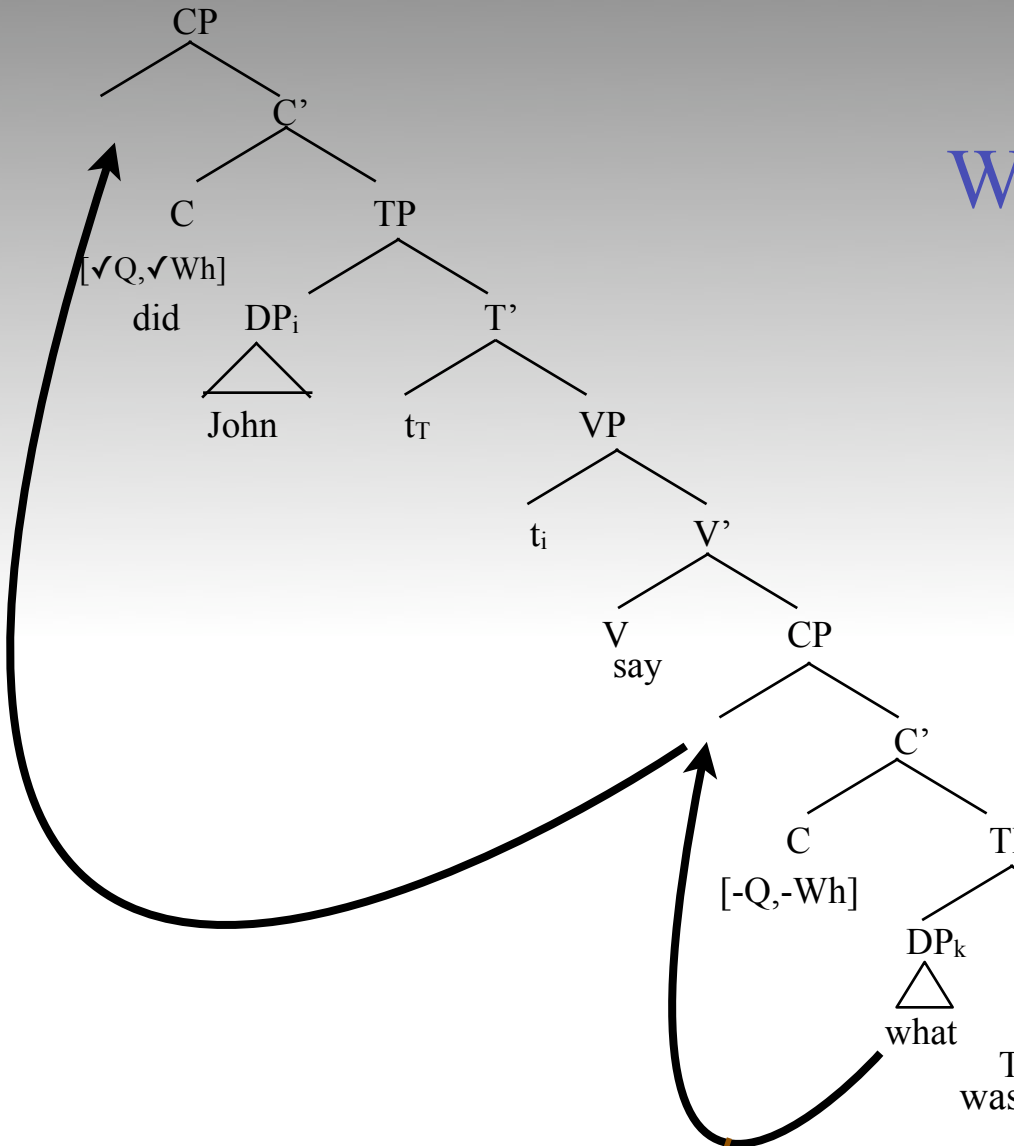


This move is mysterious for the moment, but will be explained shortly

Wh-movement

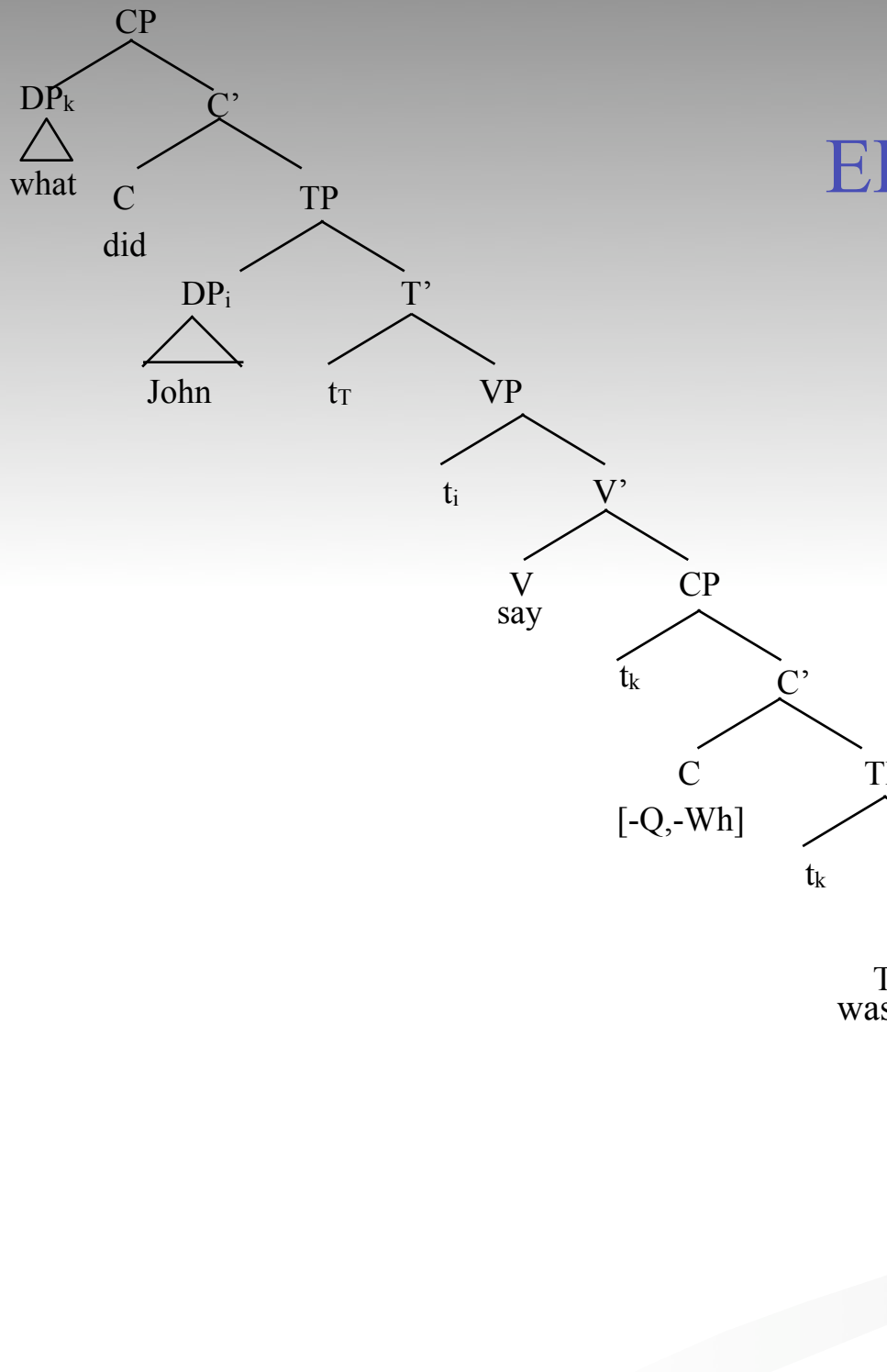
Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- ✓ •Requirement that features are checked
 - ✓•[past]
 - ✓•[+Q]
 - ✓•[+Wh]
- EPP



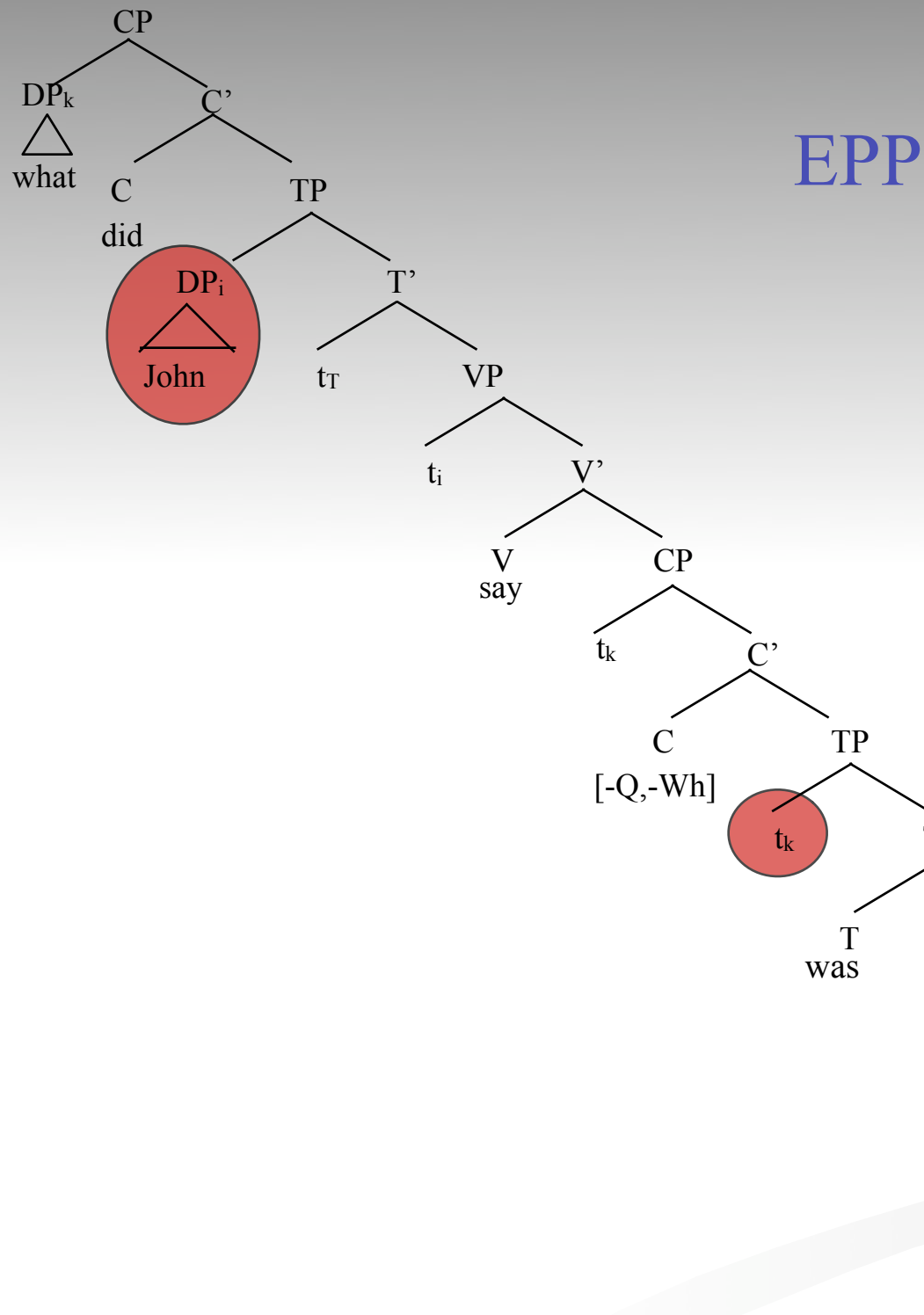
This move is mysterious for the moment, but will be explained shortly

EPP



Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- ✓ •Requirement that features are checked
 - ✓•[past]
 - ✓•[+Q]
 - ✓•[+Wh]
- EPP



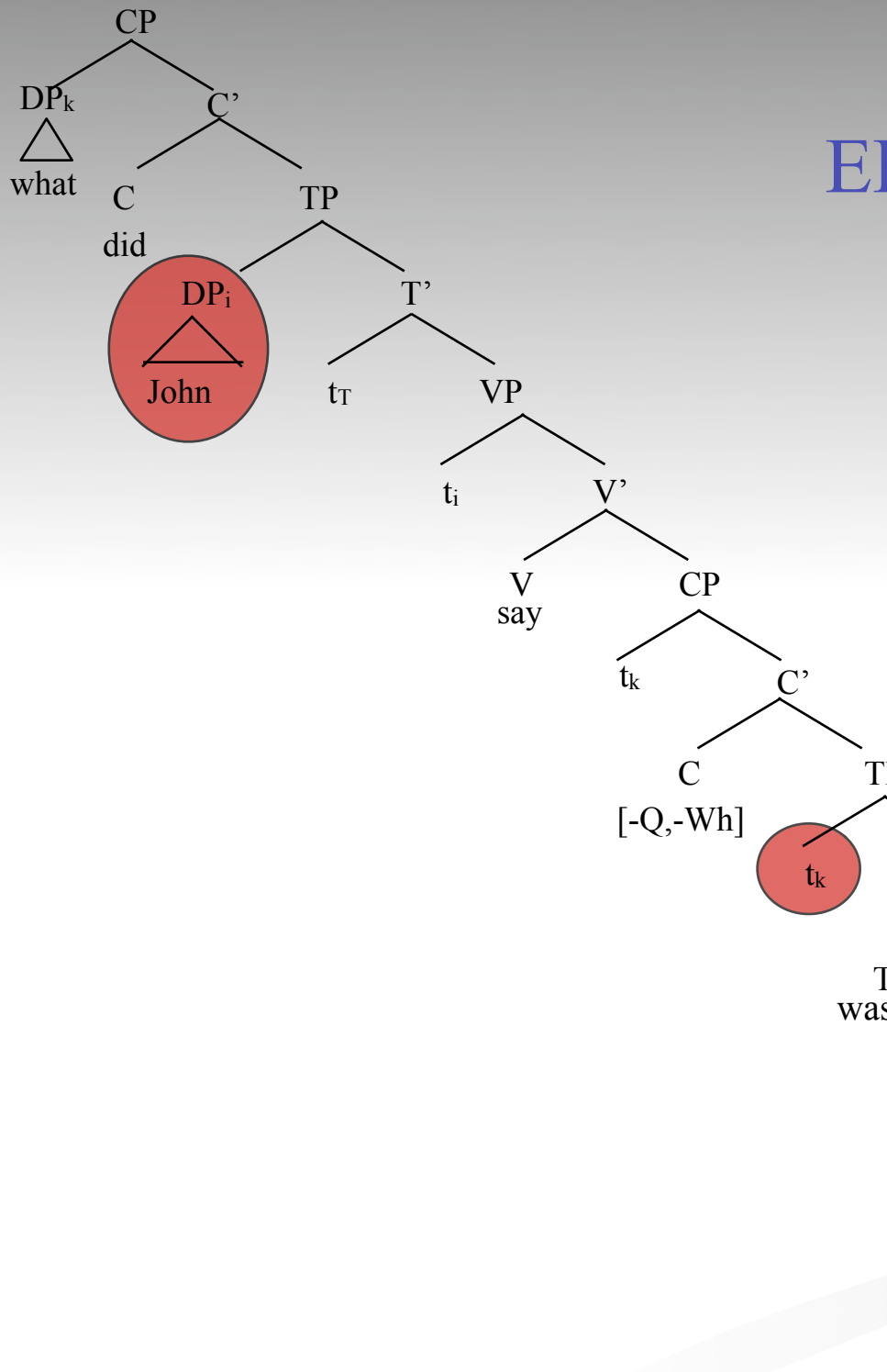
EPP

Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- ✓ •Requirement that features are checked
 - ✓•[past]
 - ✓•[+Q]
 - ✓•[+Wh]

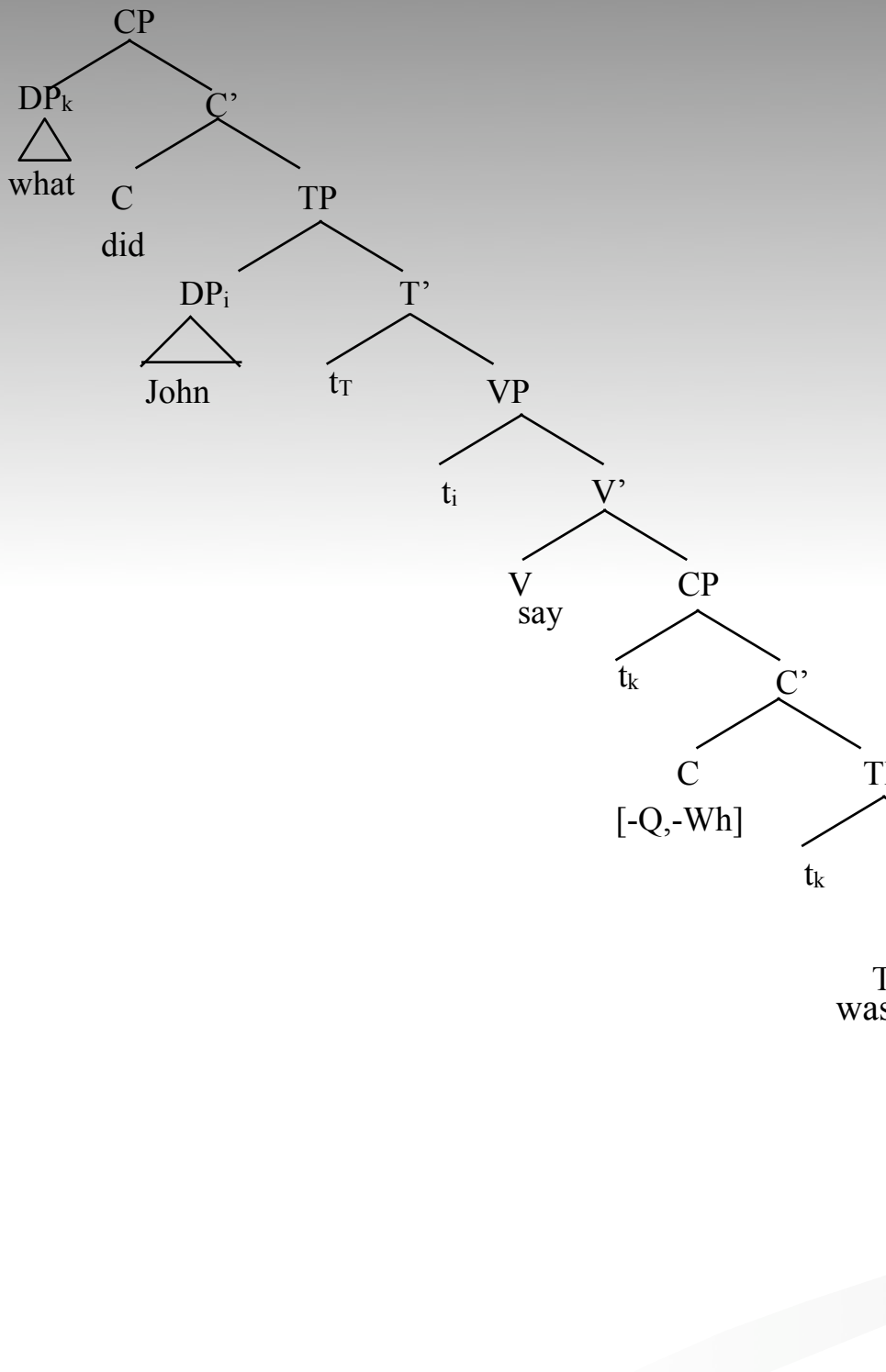
•EPP

EPP



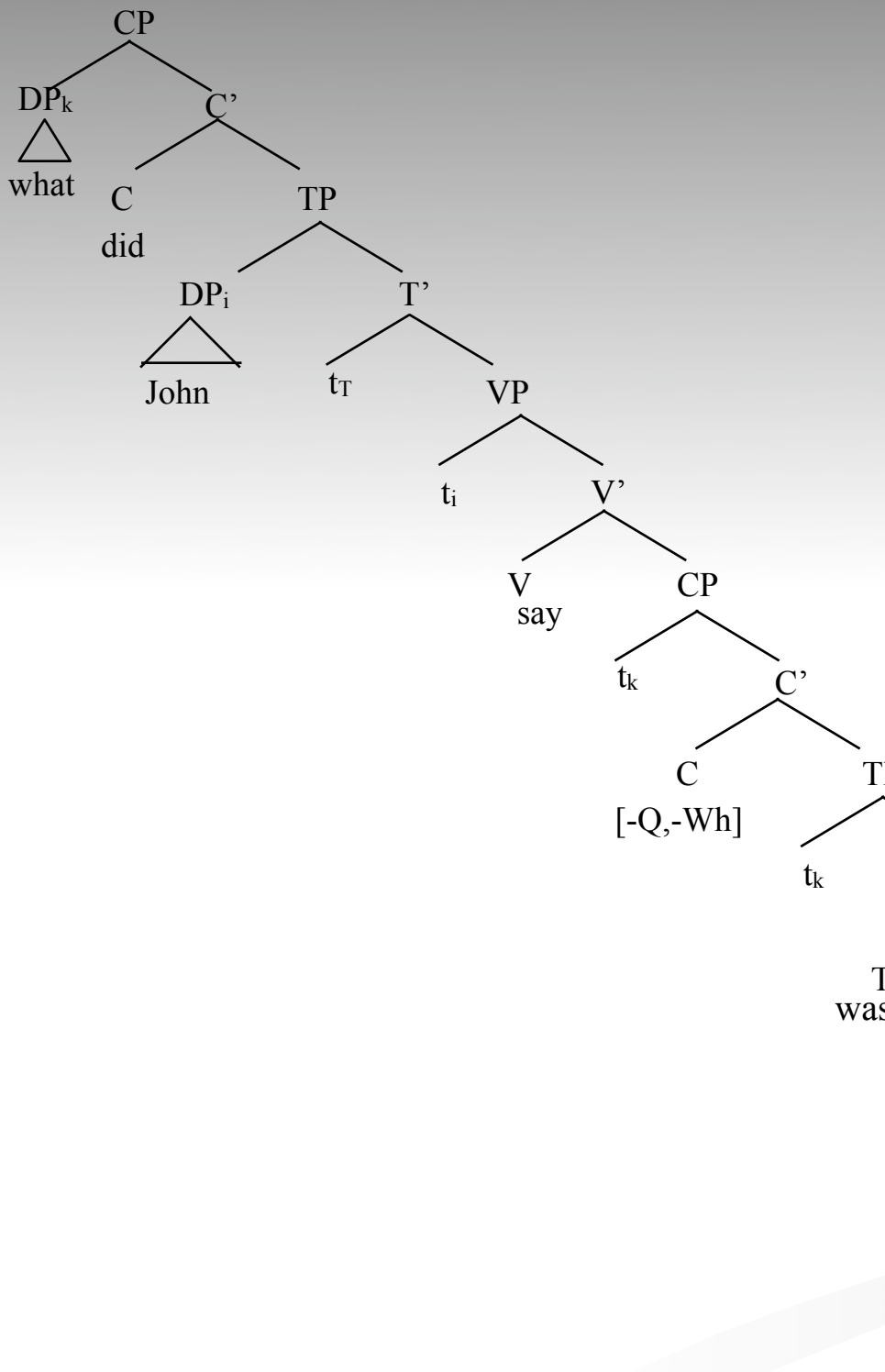
Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- ✓ •Requirement that features are checked
 - ✓•[past]
 - ✓•[+Q]
 - ✓•[+Wh]
- ✓ •EPP



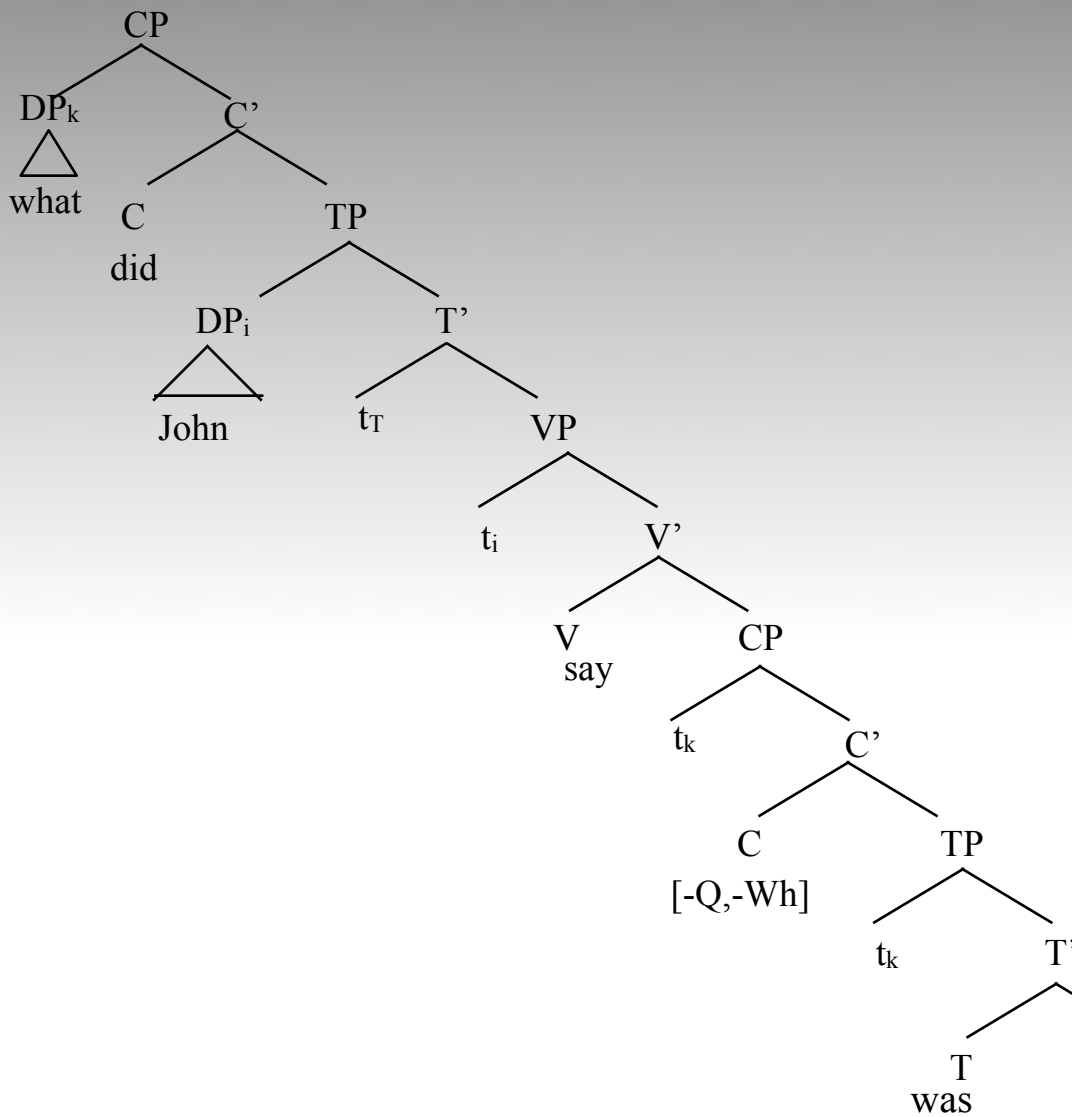
Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- ✓ •Requirement that features are checked
 - ✓•[past]
 - ✓•[+Q]
 - ✓•[+Wh]
- EPP



Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- ✓ •Requirement that features are checked
 - ✓•[past]
 - ✓•[+Q]
 - ✓•[+Wh]
- ✓ •EPP

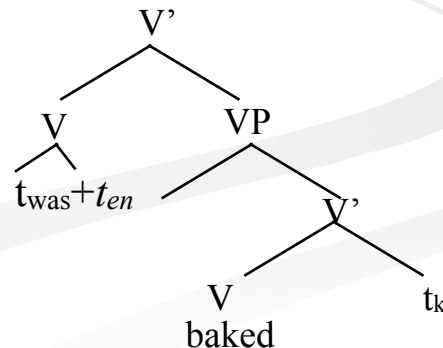


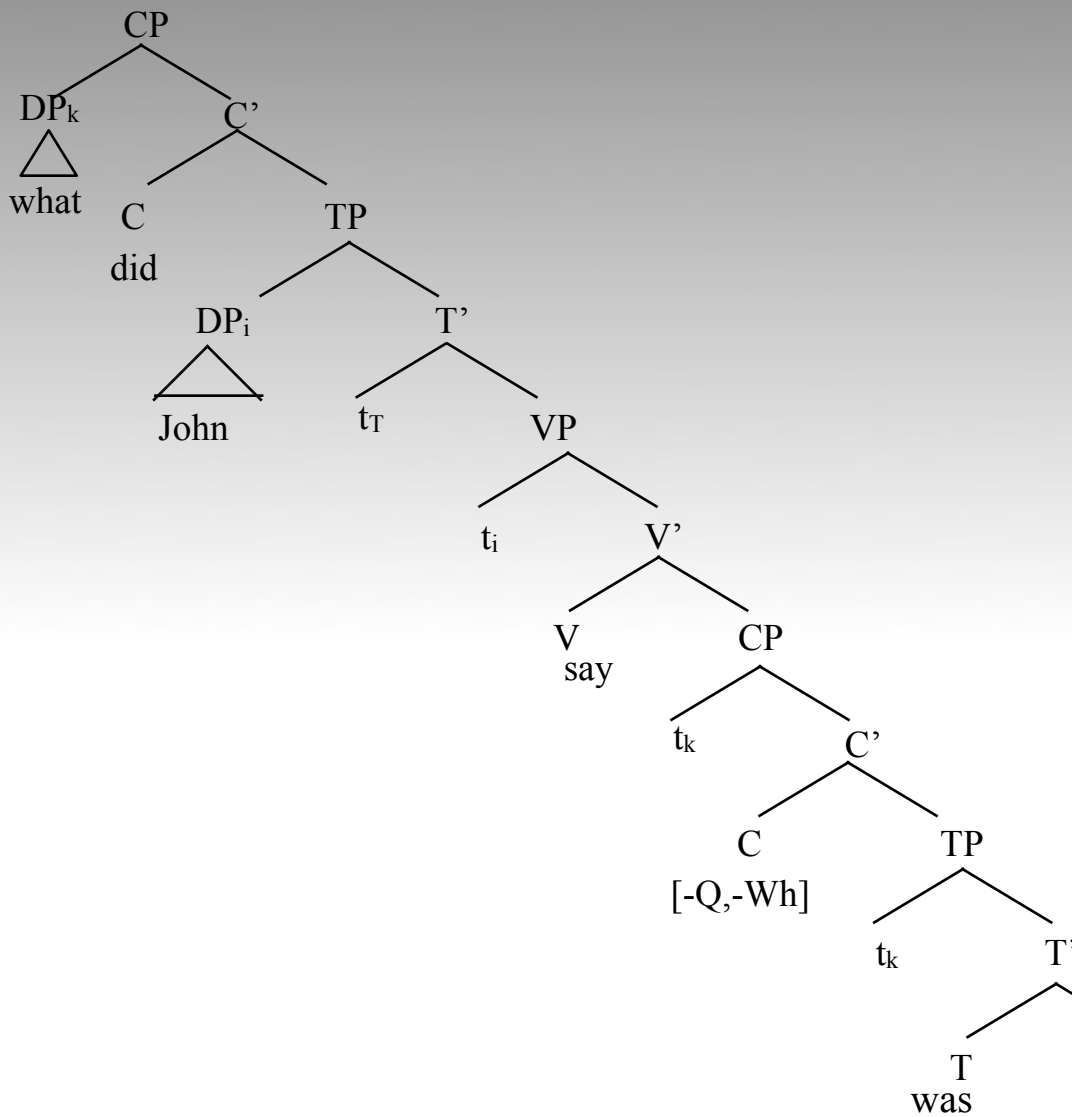
Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- ✓ •Requirement that features are checked
 - ✓•[past]
 - ✓•[+Q]
 - ✓•[+Wh]
- ✓ •EPP

English-specific Constraints to be met

- That trace filter*- cf. What did John say that t was baked
- Doubly filled CP* cf. John said what that was baked



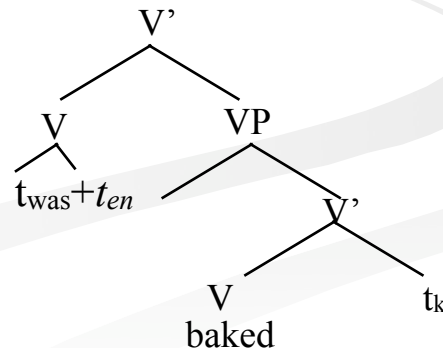


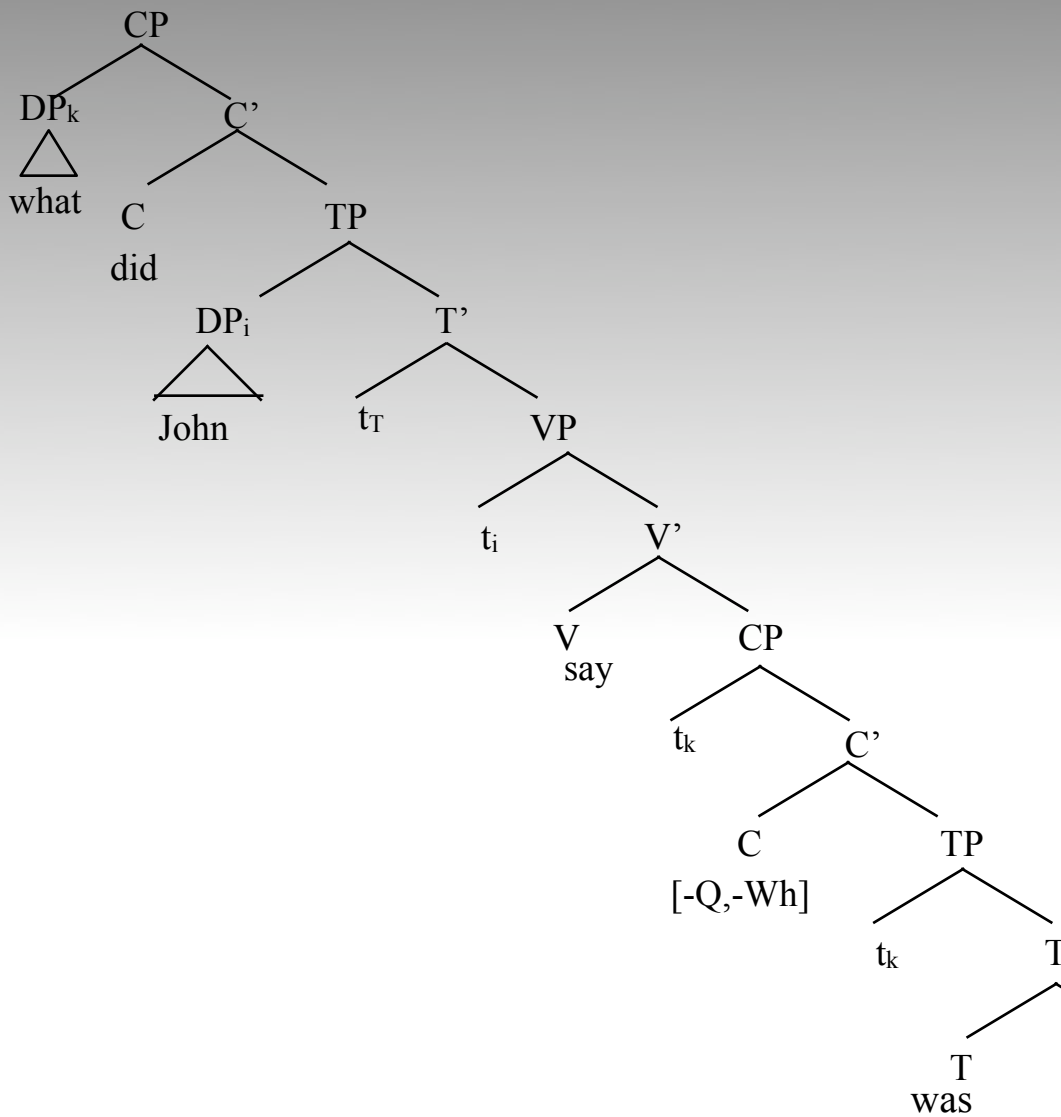
Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- ✓ •Requirement that features are checked
 - ✓•[past]
 - ✓•[+Q]
 - ✓•[+Wh]
- ✓ •EPP

English-specific Constraints to be met

- ✓ •*That trace filter*- cf. What did John say that t was baked
- Doubly filled CP* cf. John said what that was baked



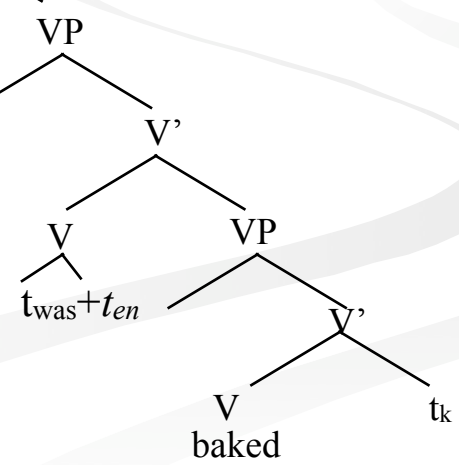


Constraints to be met

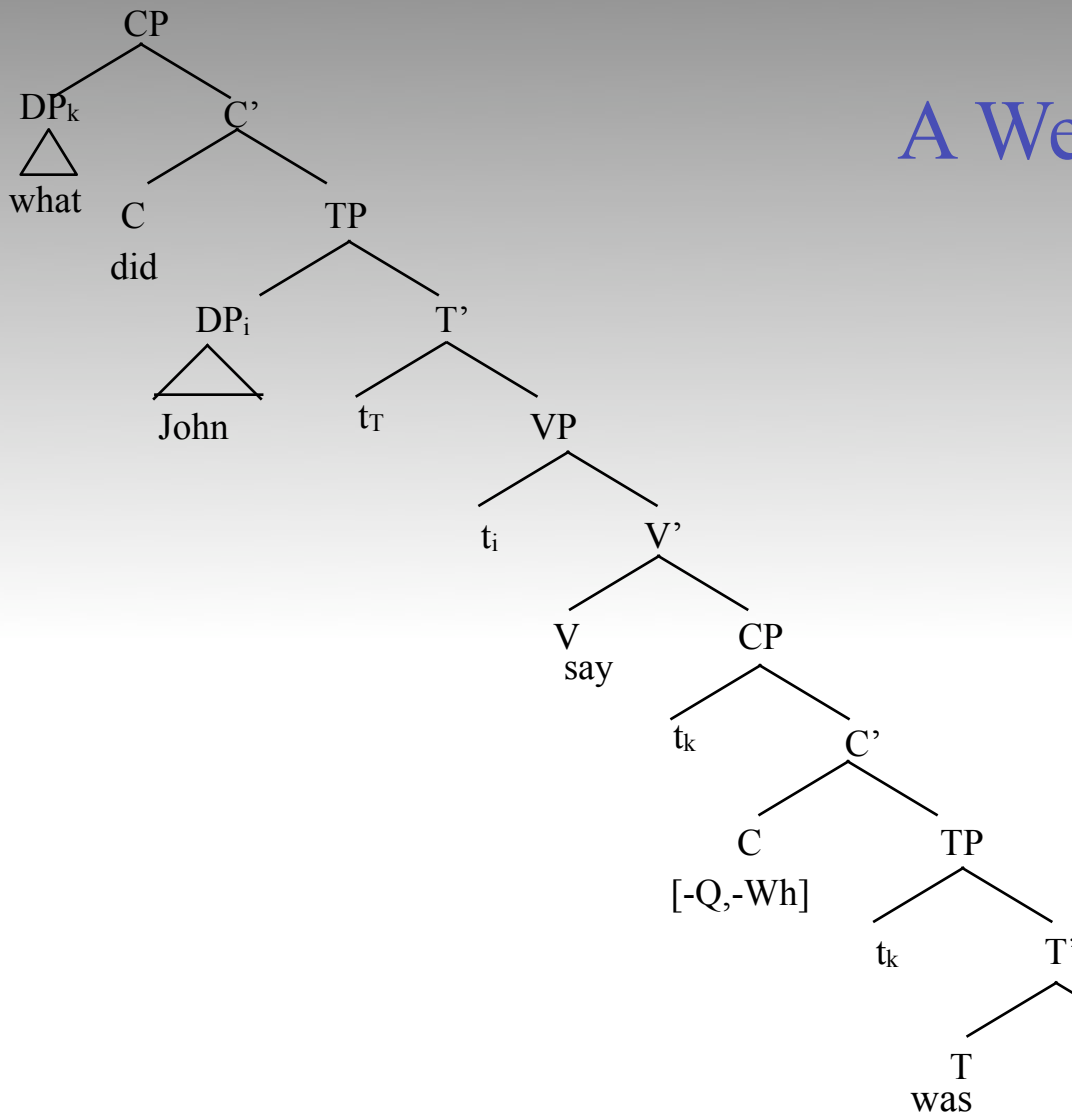
- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- ✓ •Requirement that features are checked
 - ✓•[past]
 - ✓•[+Q]
 - ✓•[+Wh]
- ✓ •EPP

English-specific Constraints to be met

- ✓ •*That trace filter*- cf. What did John say that t was baked
- ✓ •*Doubly filled CP* cf. John said what that was baked



A Well-formed S-structure

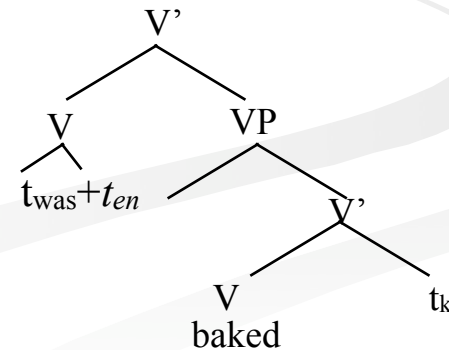


Constraints to be met

- ✓ •Theta Criterion
- ✓ •Case Filter
- ✓ •Requirement that suffixes are attached
- ✓ •Requirement that features are checked
 - ✓•[past]
 - ✓•[+Q]
 - ✓•[+Wh]
- ✓ •EPP

English-specific Constraints to be met

- ✓ •*That trace filter*- cf. What did John say that t was baked
- ✓ •*Doubly filled CP* cf. John said what that was baked



Traces?

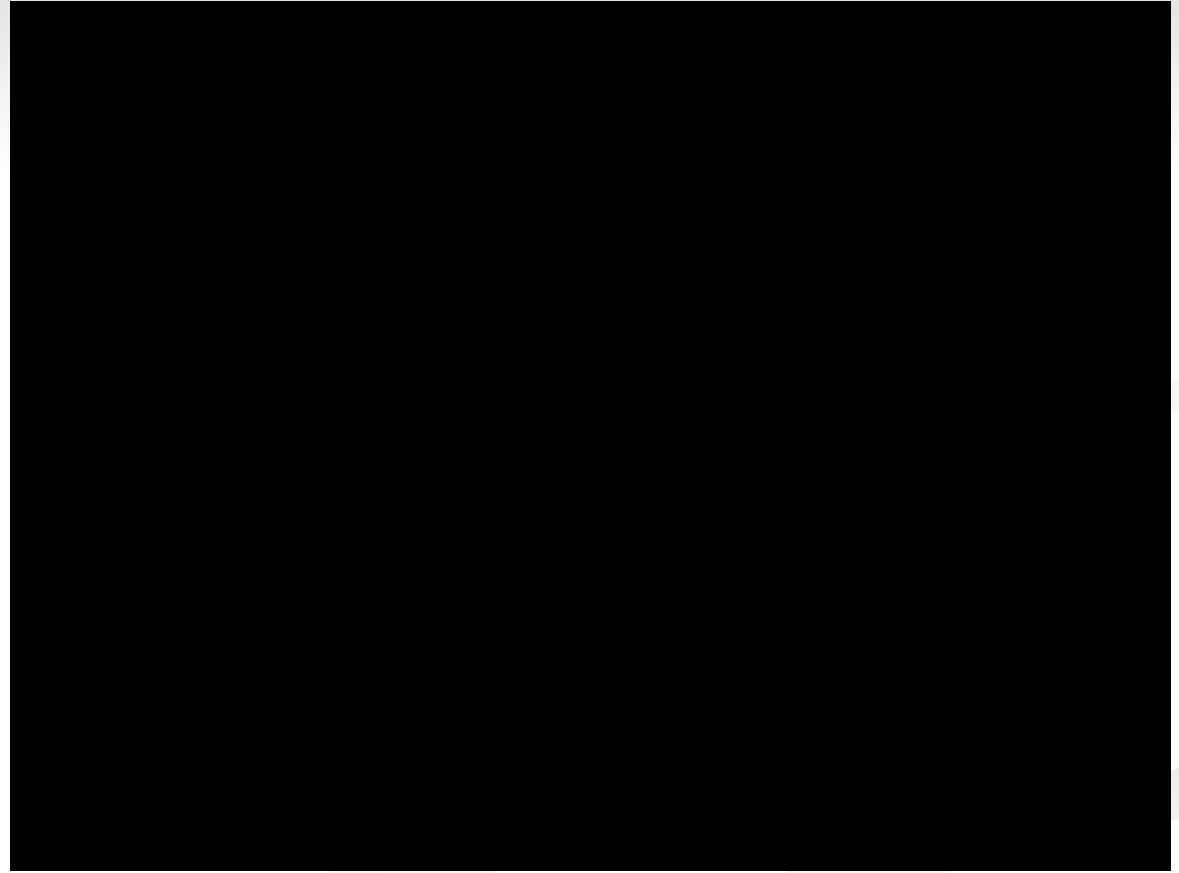
- want + to → wanna
 - Who_i do you wanna kiss *t_i*?
 - *Who do you wanna kiss the puppy?
 - Who_i do you want *t_i* to kiss the puppy?

Traces?

- want + to → wanna
 - Who_i do you wanna kiss t_i?
 - *Who do you wanna kiss the puppy?
 - Who_i do you want t_i to kiss the puppy?

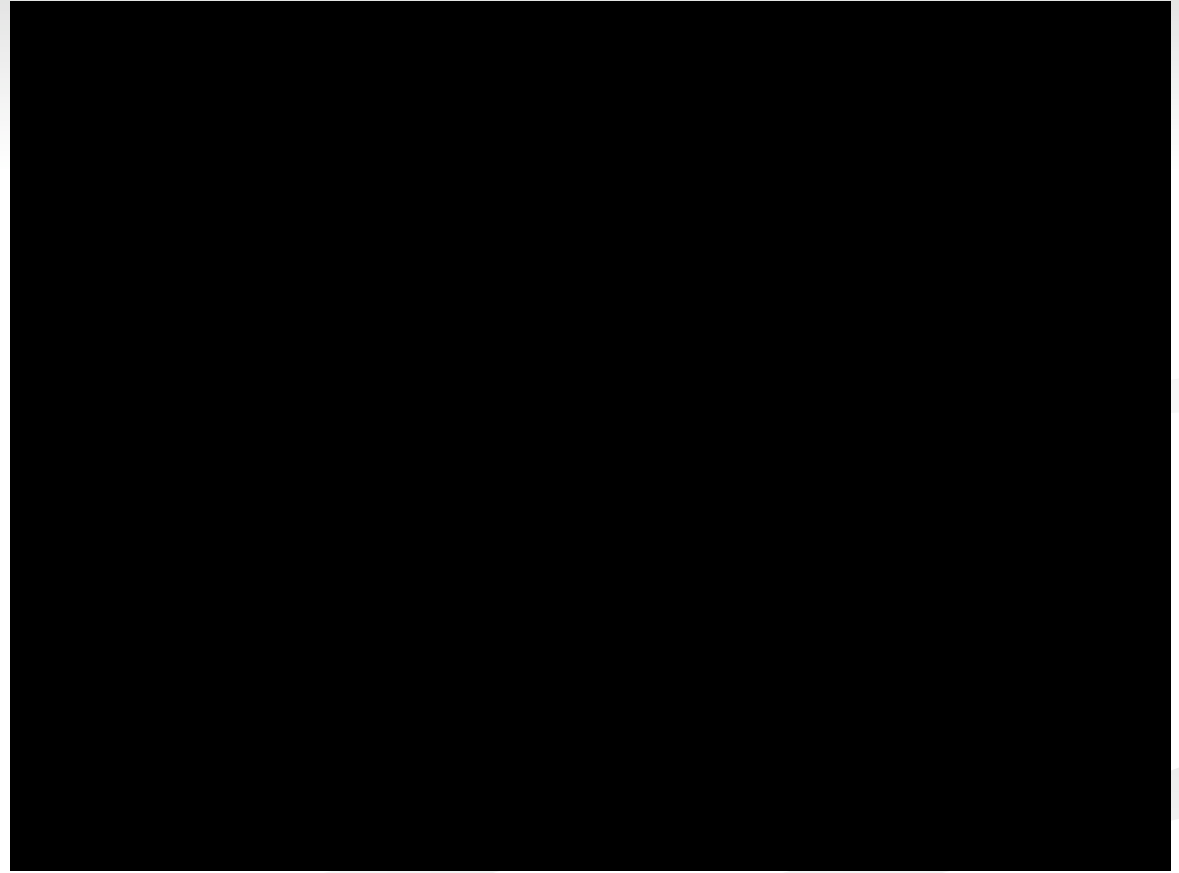
↑
intervenes, so blocks wanna
contraction

Children and Traces



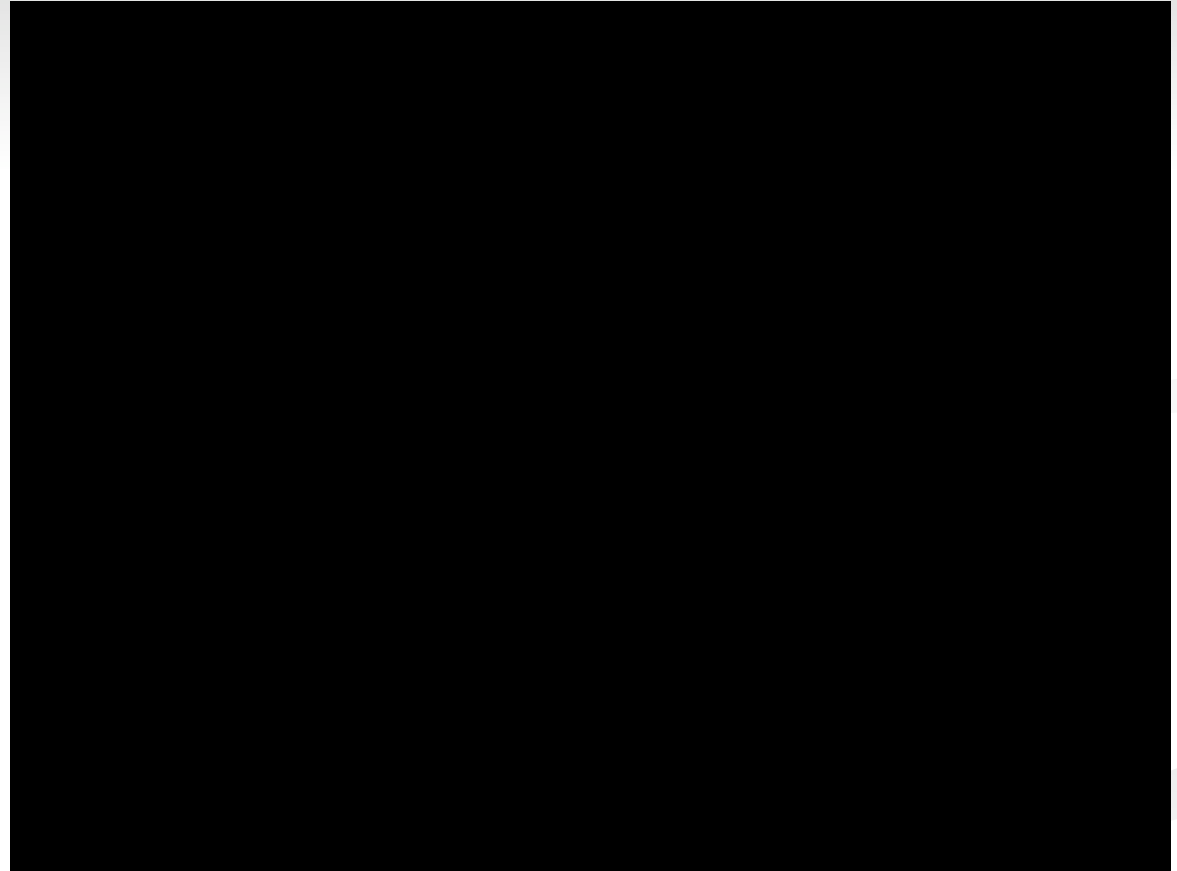
Children and Traces

- This movie shows three things:



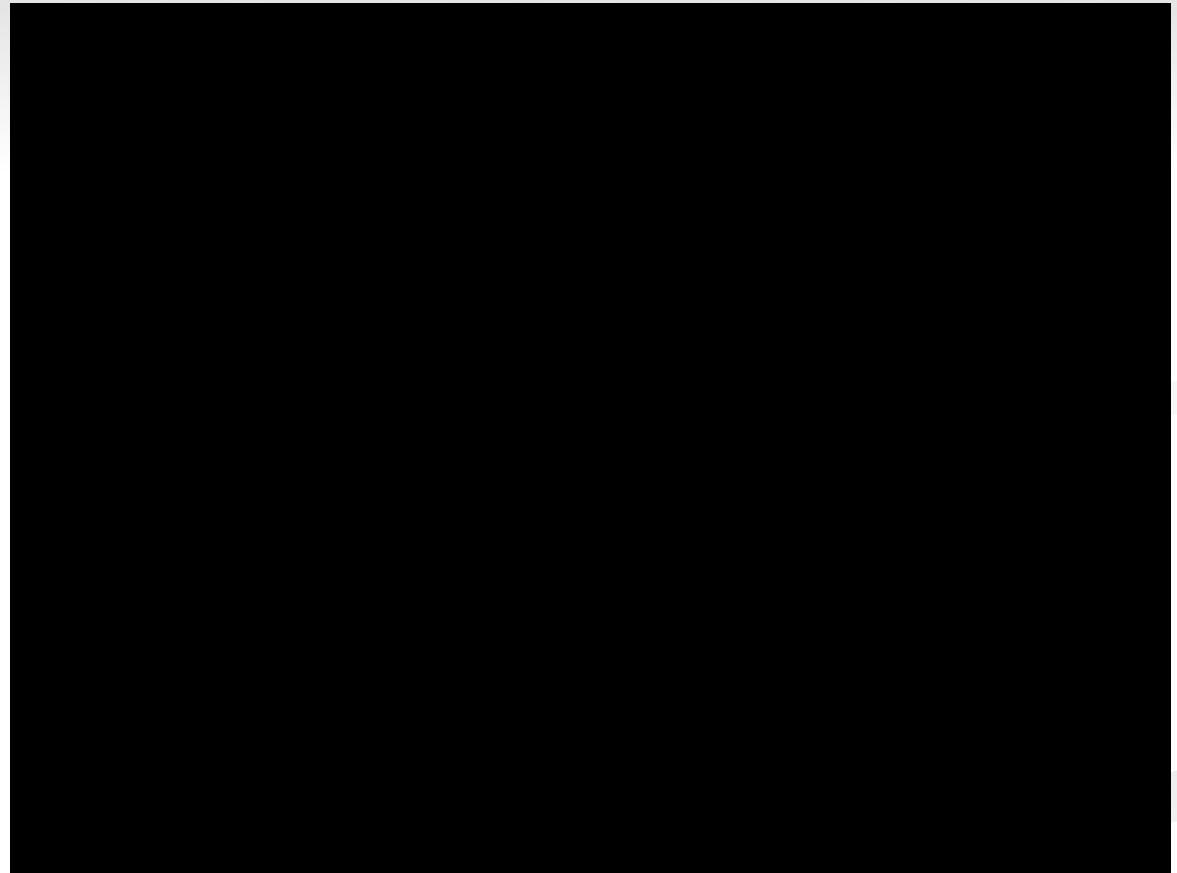
Children and Traces

- This movie shows three things:
- -Children exhibiting *overt* traces (saying the word in both its D-structure and S-structure position)



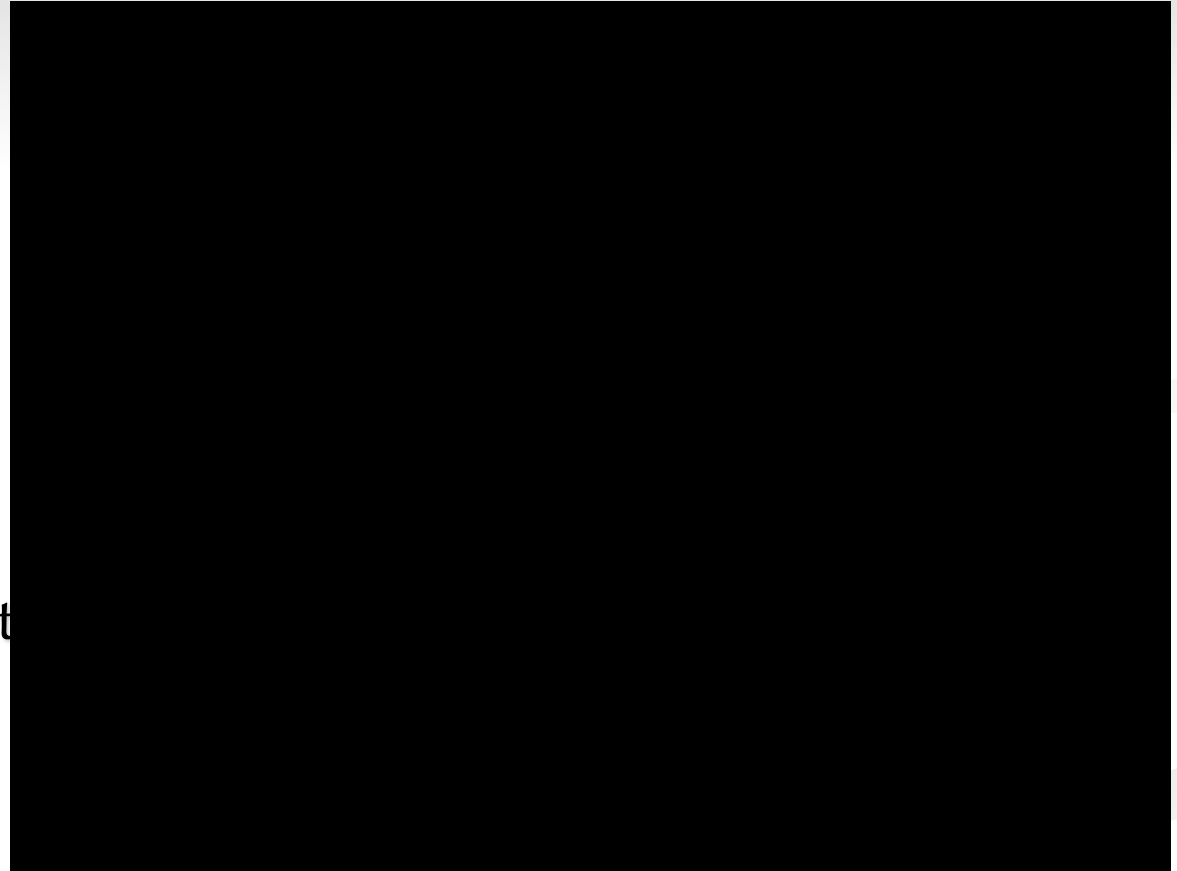
Children and Traces

- This movie shows three things:
- -Children exhibiting *overt* traces (saying the word in both its D-structure and S-structure position)
- Children never pronounce a wh-word in a *wanna* contraction environment



Children and Traces

- This movie shows three things:
- -Children exhibiting *overt* traces (saying the word in both its D-structure and S-structure position)
- Children never pronounce a wh-word in a *wanna* contraction environment
- Children don't *wanna* contract across a wh-trace.



What summary?

- Wh-movement moves Wh-phrase to specifier of CP
- Motivated by need to get Wh-phrase near [+WH] complementizer
- Two English Specific constraints
 - Doubly filled Comp Filter
 - That-trace filter