Syntactic constraints: Strings are more insightful than trees

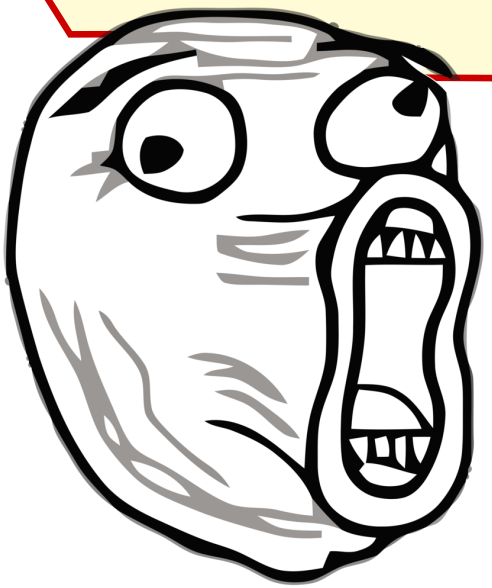
Thomas Graf

Stony Brook University
mail@thomasgraf.net

Department of Linguistics
University of Rochester
September 11, 2026

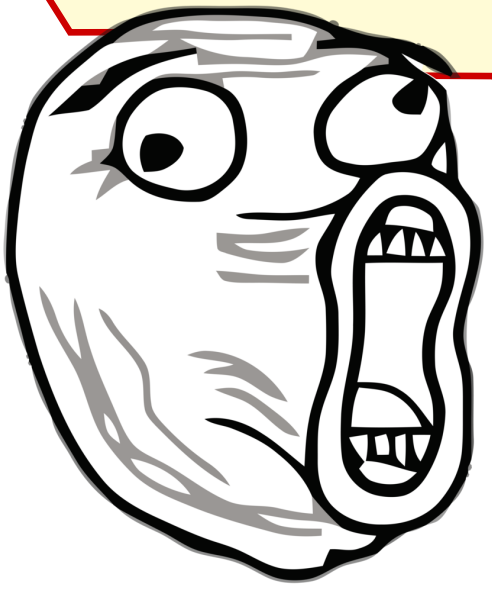
The talk in a nutshell

lol trees are so 1990 what
r linguists evne dooing u
only need sstringws u dummies!!!1!!!11!



The talk in a nutshell

lol trees are so 1990 what
r linguists evne dooing u
only need sstringws u dummies!!!1!!!11!



But now that I've rage baited you into paying attention,
let's be (just a bit) more serious.

The talk in a nutshell [real version]

- Syntax may well build trees.
  Perhaps due to interface demands?
- **But:** Syntactic computations don't exploit the power of trees. They operate on strings that track certain **derivational paths**.
- Why does that matter?
  - Attested syntactic constraints form a **natural class**.
    movement, islands, relativized minimality, binding of reflexives, extraction morphology, freezing, ...
  - Surprising patterns become expected.
    typology of extraction morphology, German wh-copying
  - Learnability
  - Parallels to phonology and morphology
  - Building bridges to psycho- and neurolinguistics

This will be abstract, and that's a good thing

The critical link between disciplines should come from computation, specifically, from computational models that are made explicit at the appropriate level of **abstraction** *to create an interface for linguistics and neurobiology.*

(Poeppel and Embick 2005)

David Poeppel

David Embick

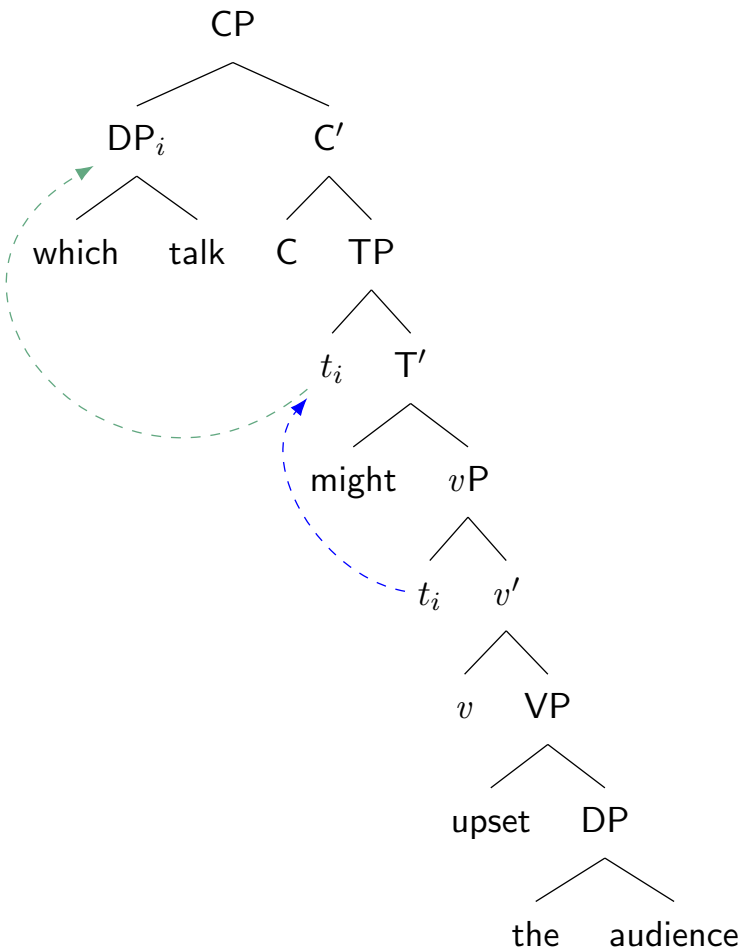
Outline

Syntax: Computation, not structures

Which talk might upset the audience?

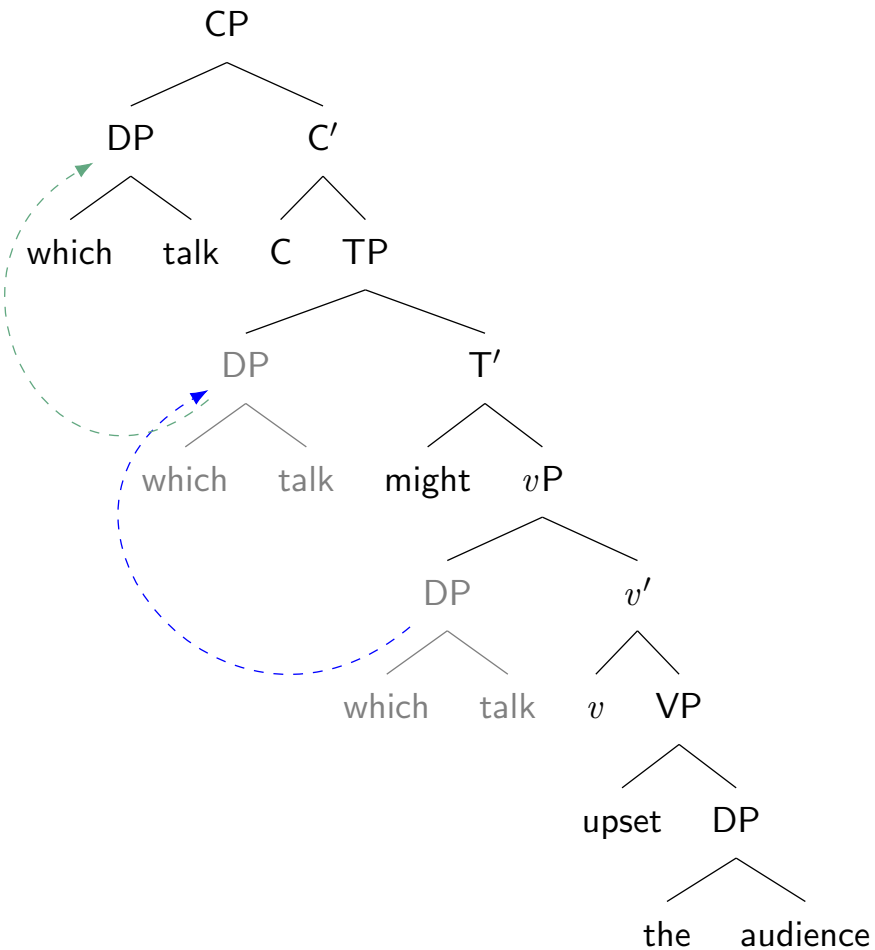
Syntax: Computation, not structures

Which talk might upset the audience?



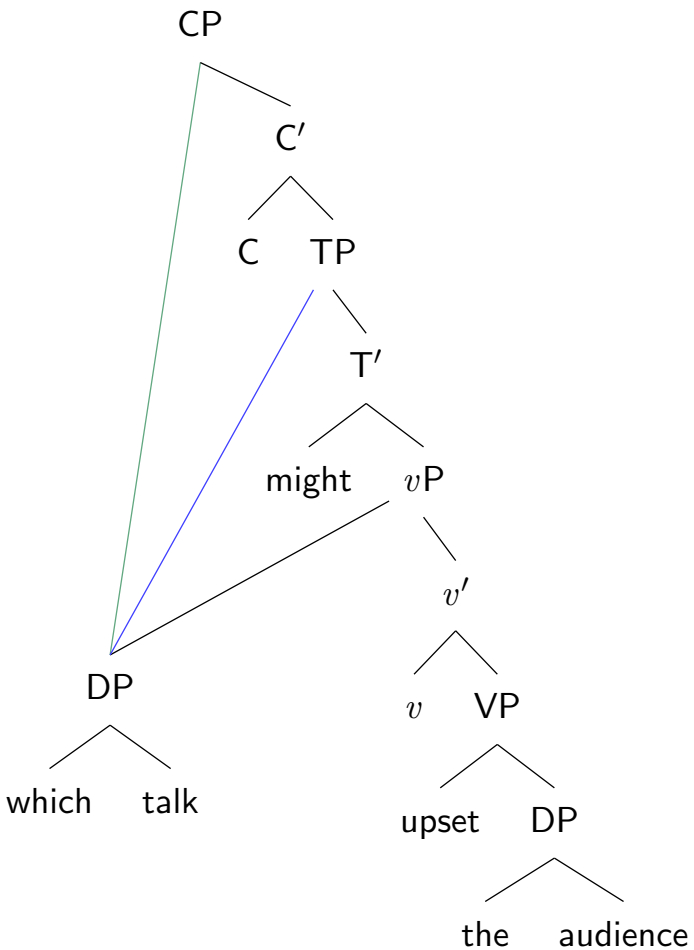
Syntax: Computation, not structures

Which talk might upset the audience?



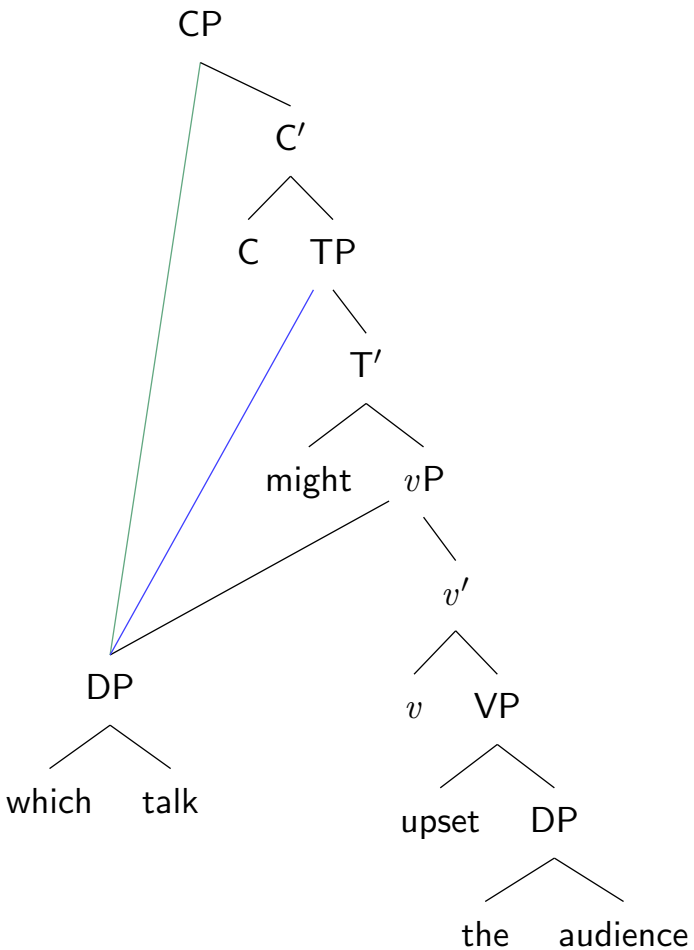
Syntax: Computation, not structures

Which talk might upset the audience?



Syntax: Computation, not structures

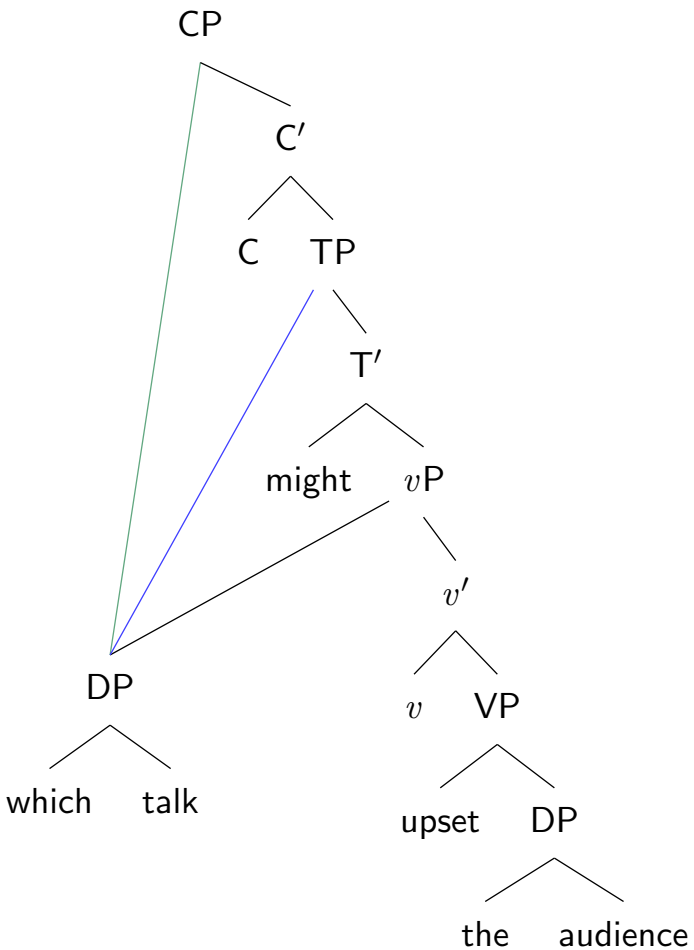
Which talk might upset the audience?

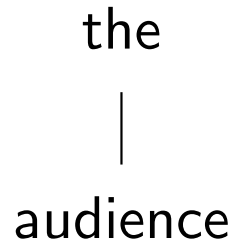


the audience

Syntax: Computation, not structures

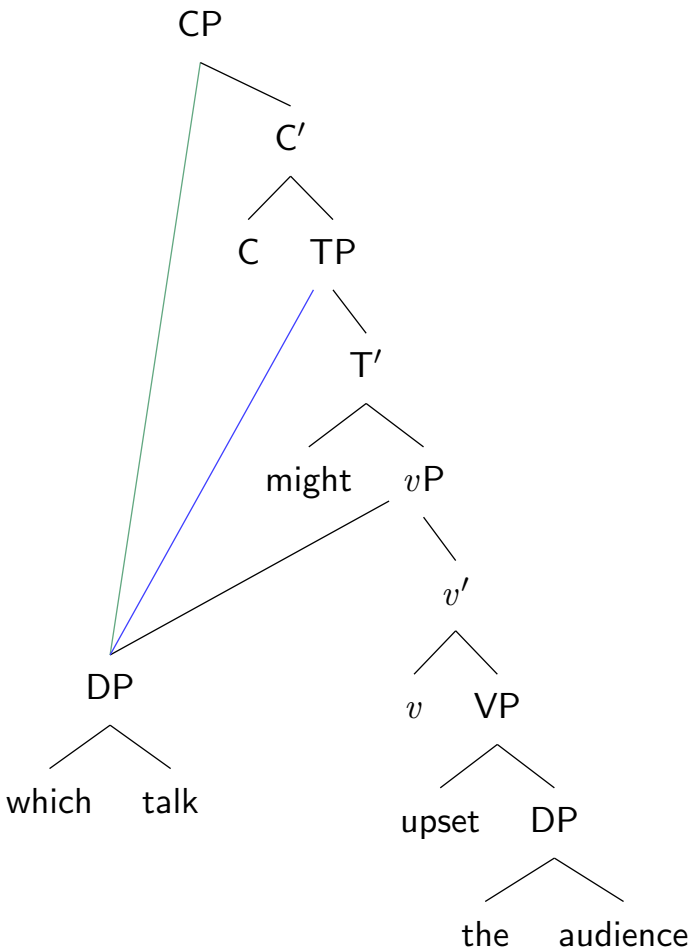
Which talk might upset the audience?

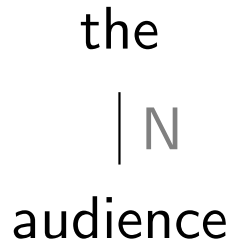




Syntax: Computation, not structures

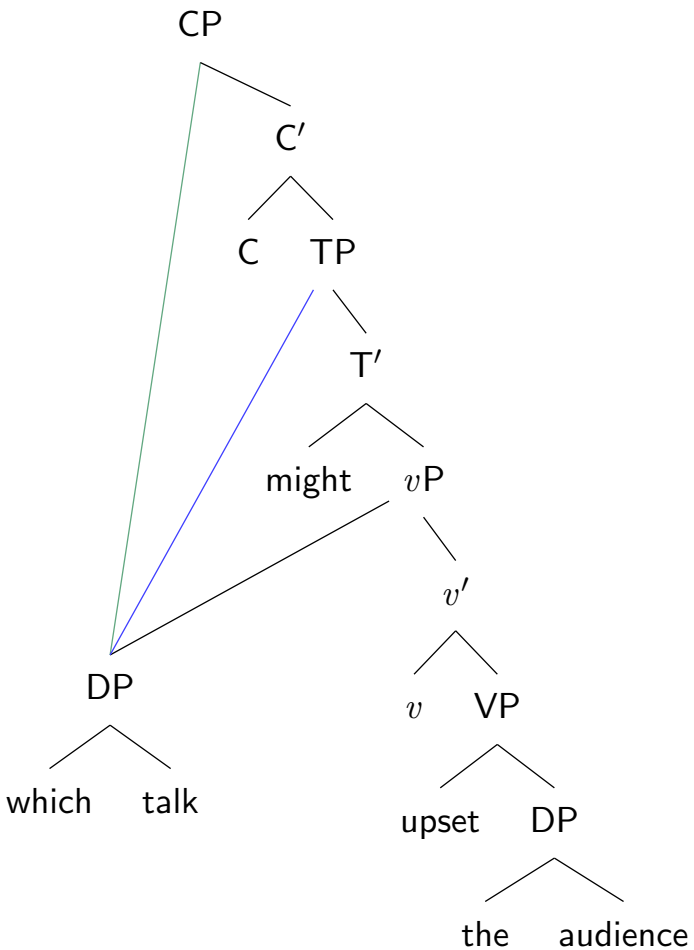
Which talk might upset the audience?





Syntax: Computation, not structures

Which talk might upset the audience?



upset

|

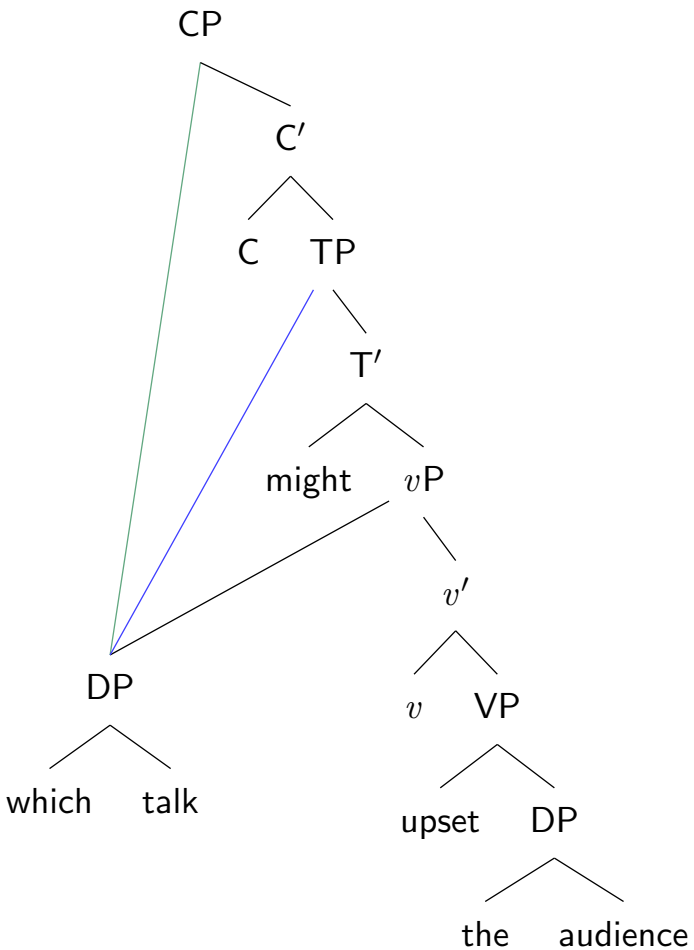
the

| N

audience

Syntax: Computation, not structures

Which talk might upset the audience?



upset

D

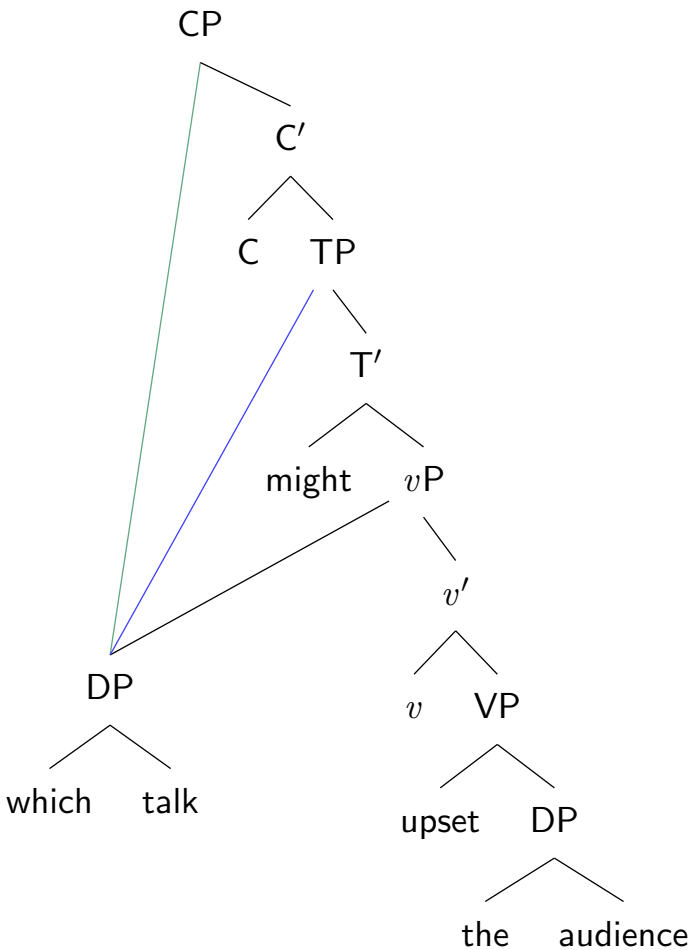
the

N

audience

Syntax: Computation, not structures

Which talk might upset the audience?

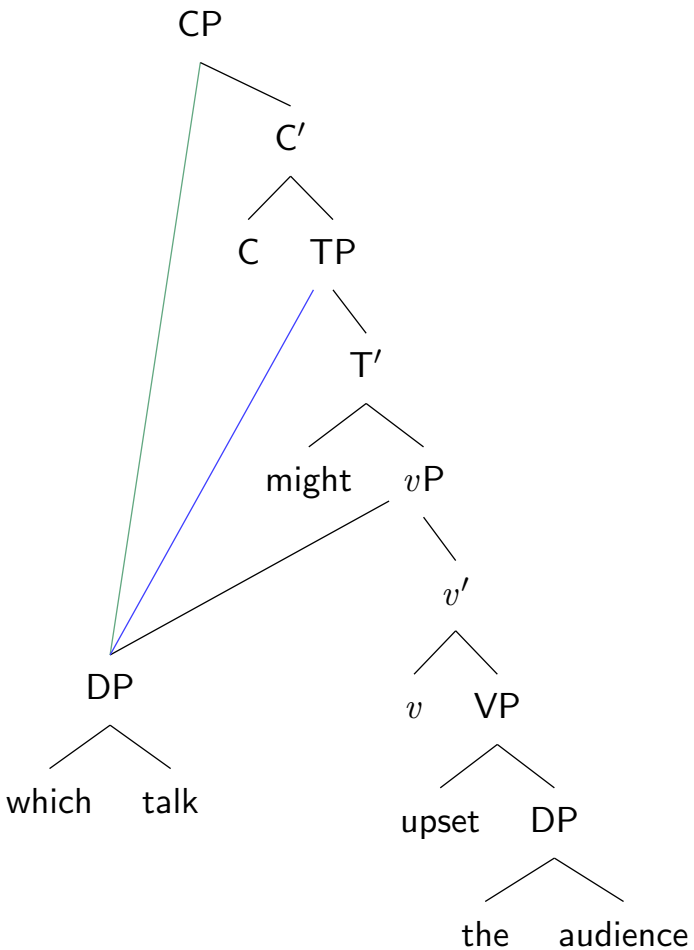


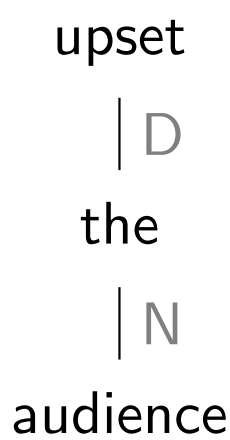
talk

upset
D
the
N
audience

Syntax: Computation, not structures

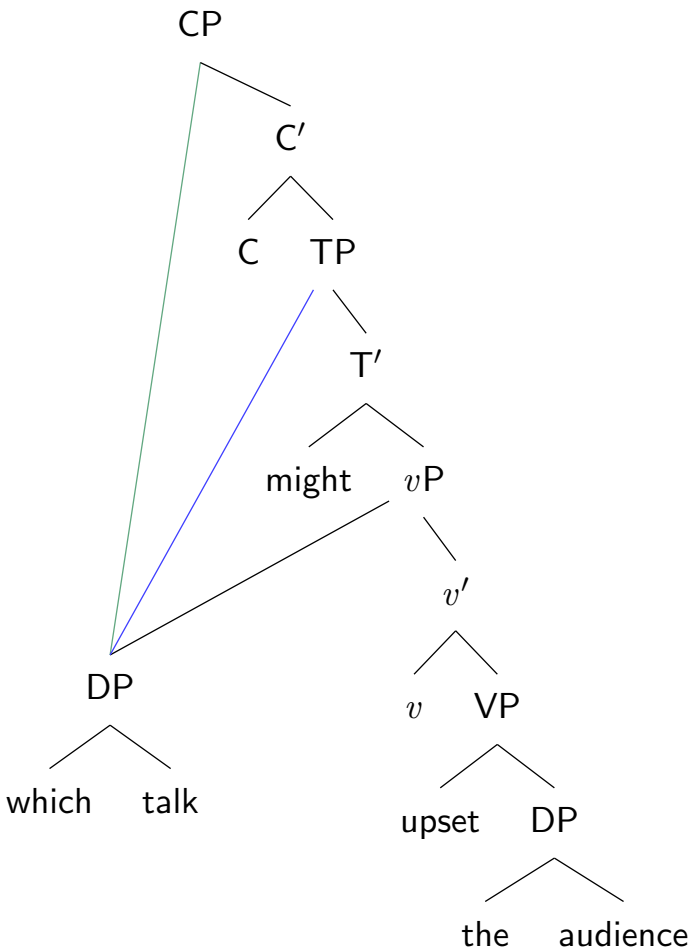
Which talk might upset the audience?

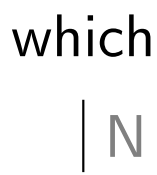


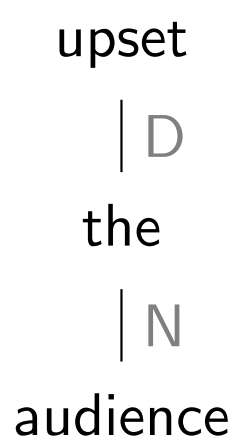


Syntax: Computation, not structures

Which talk might upset the audience?

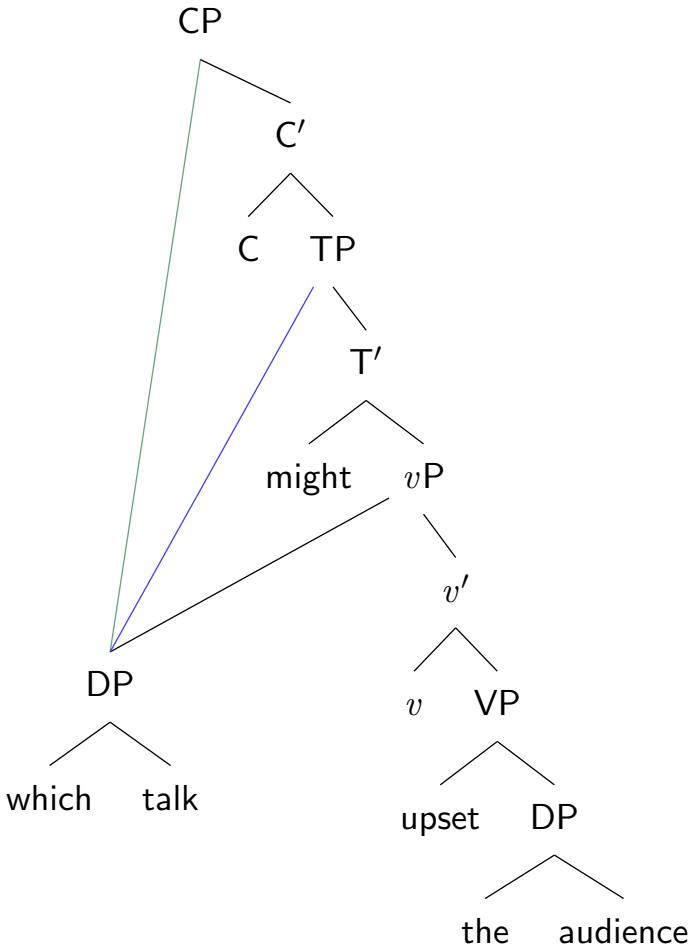


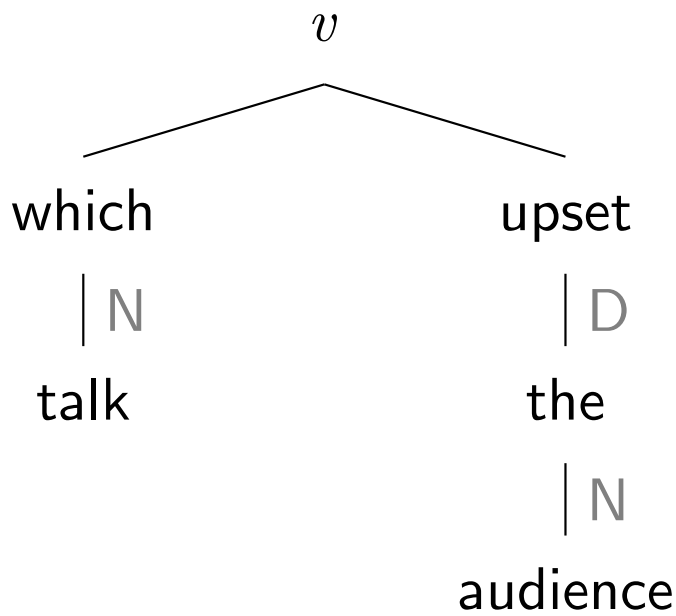




Syntax: Computation, not structures

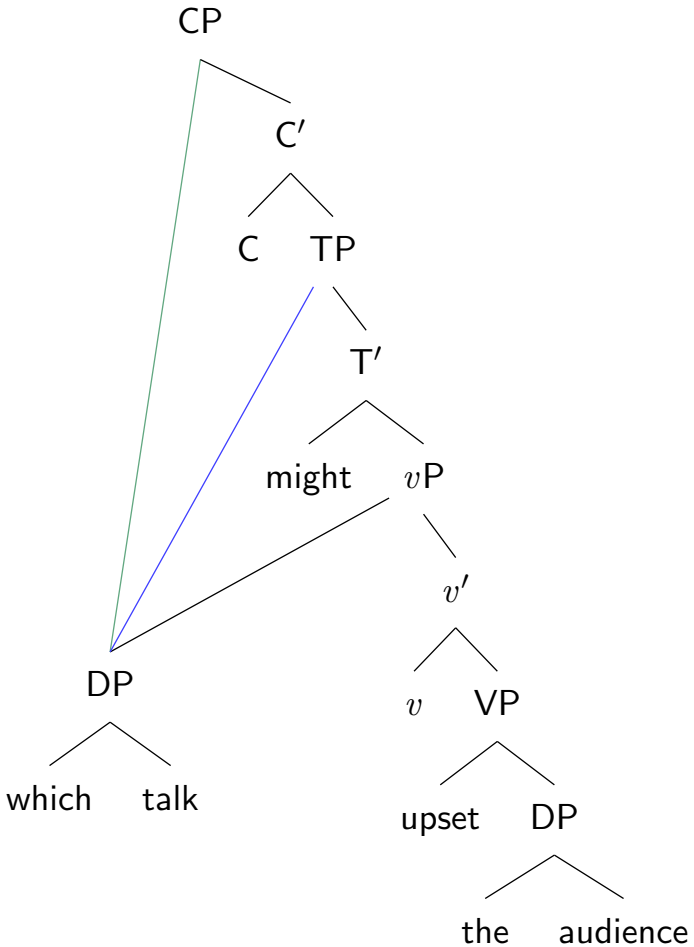
Which talk might upset the audience?

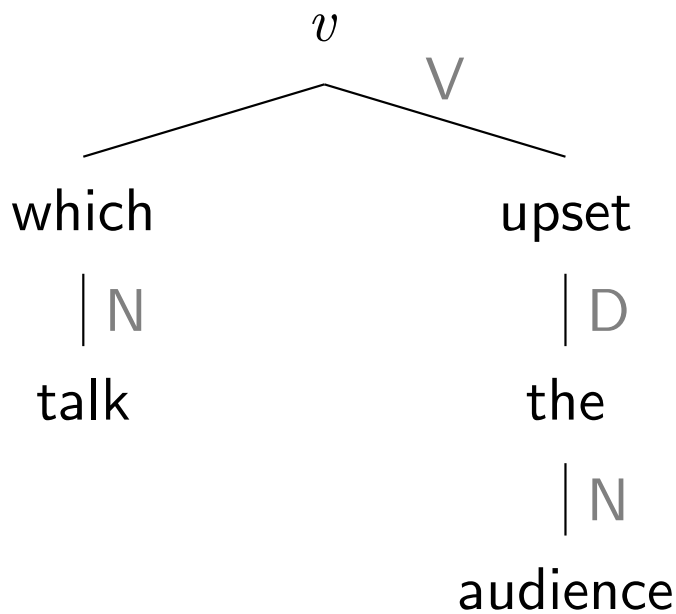




Syntax: Computation, not structures

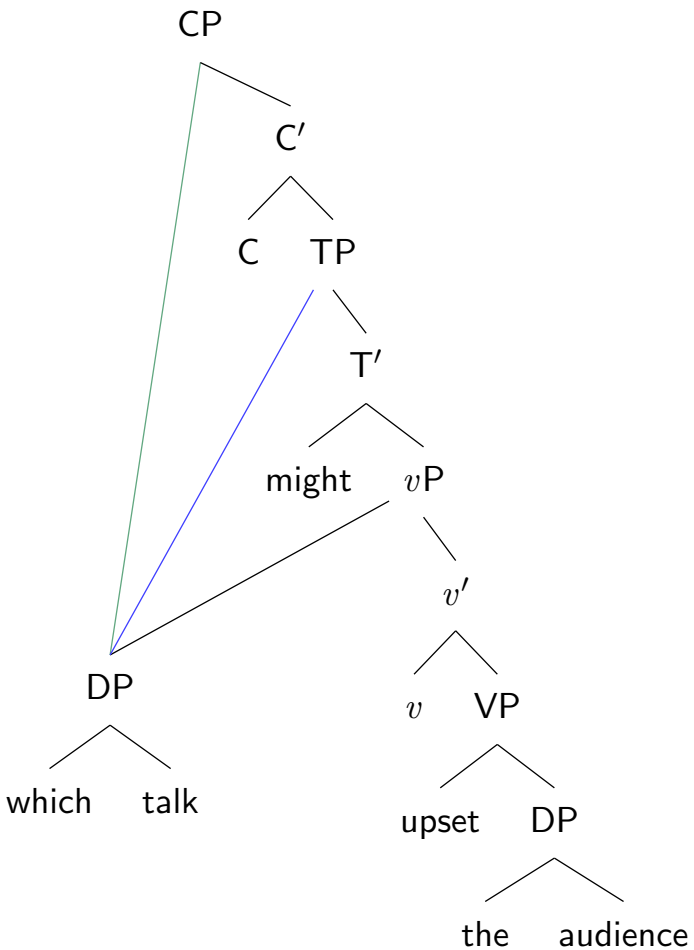
Which talk might upset the audience?

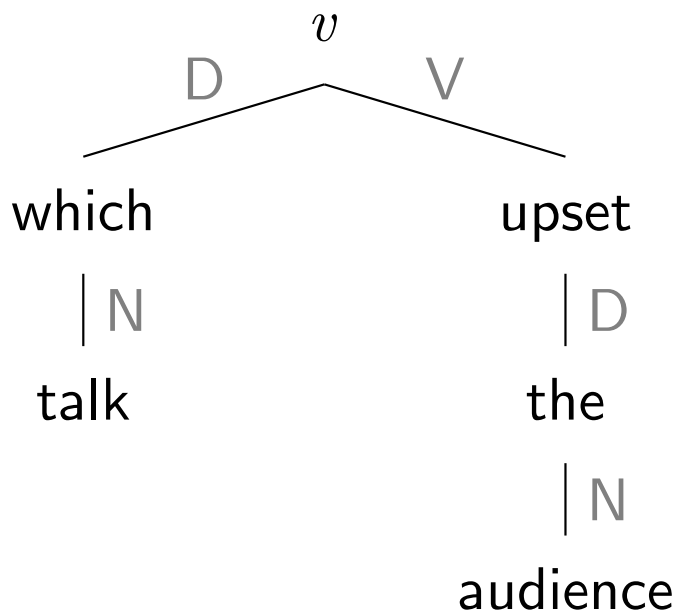




Syntax: Computation, not structures

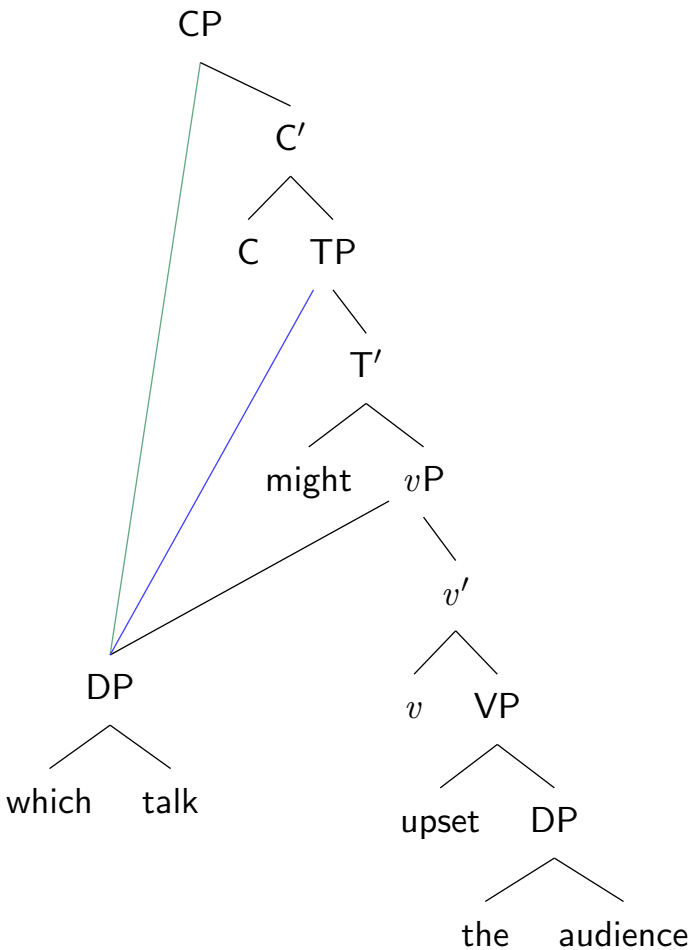
Which talk might upset the audience?

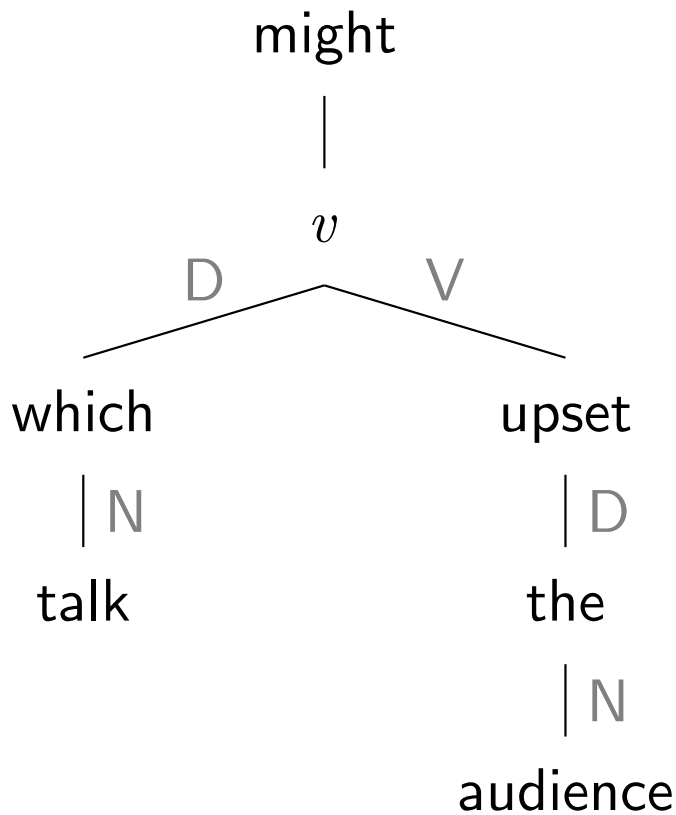




Syntax: Computation, not structures

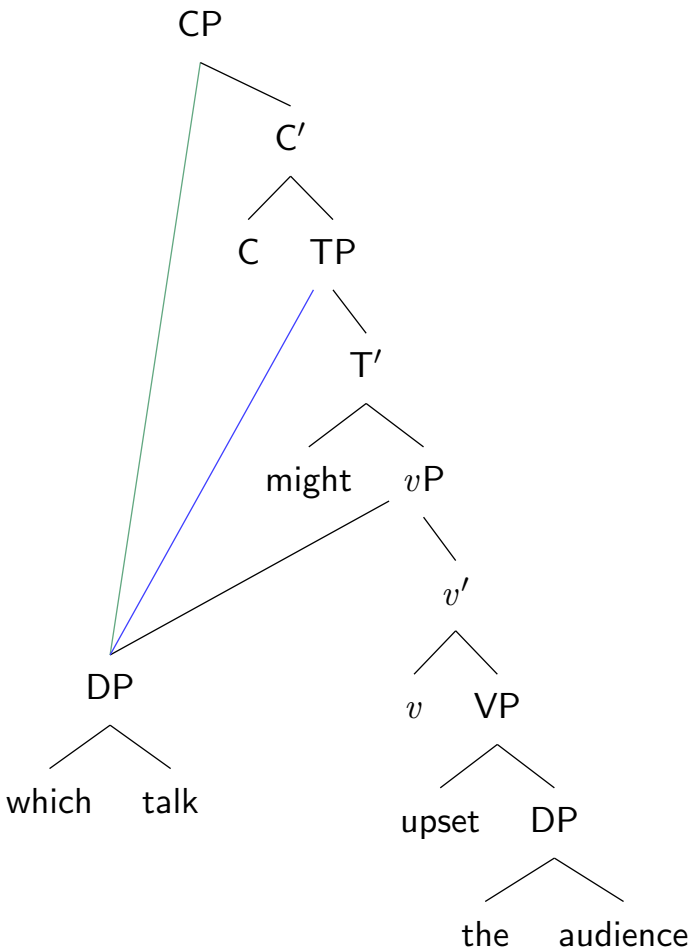
Which talk might upset the audience?

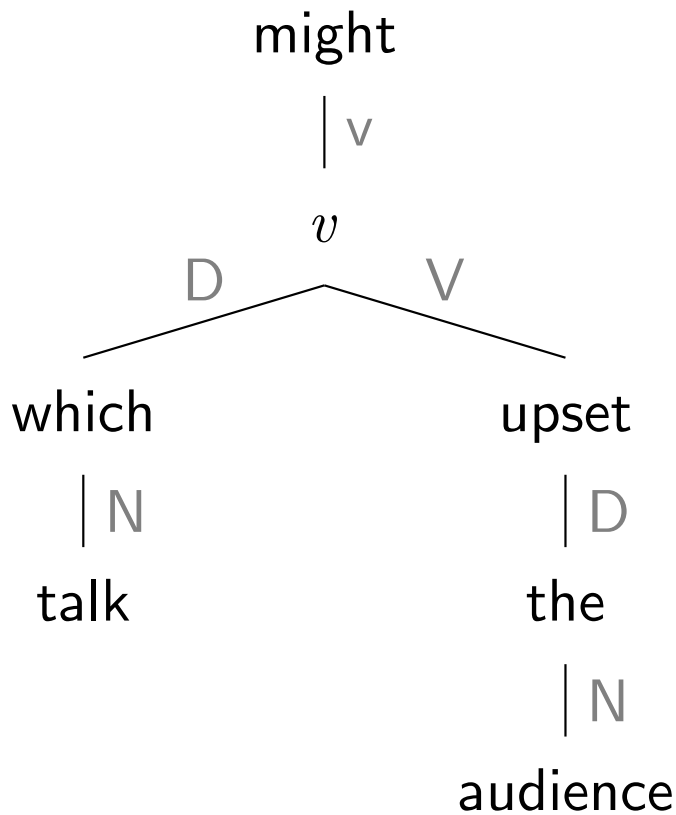




Syntax: Computation, not structures

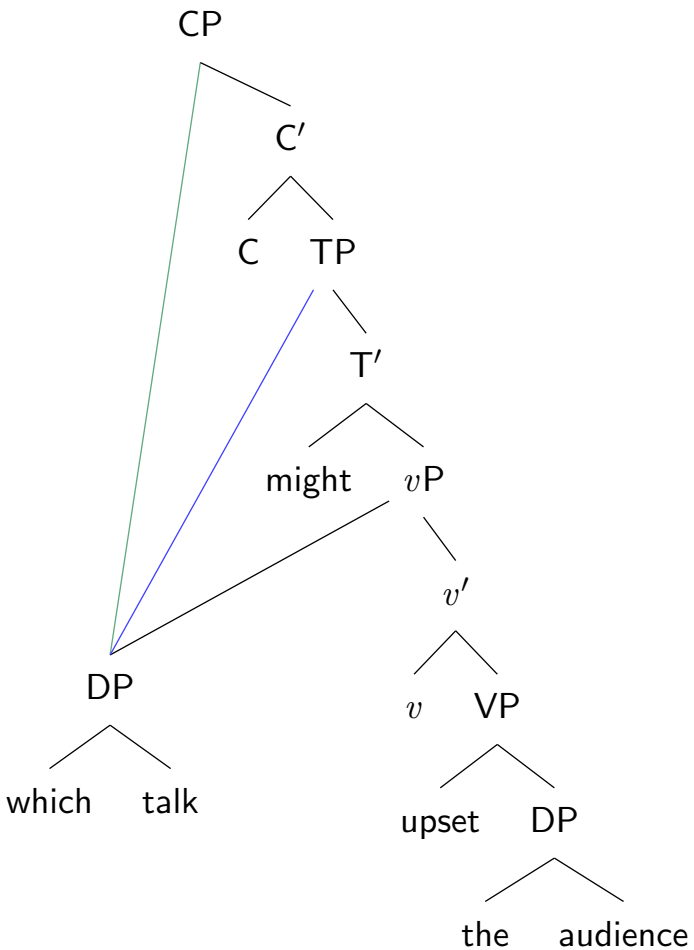
Which talk might upset the audience?

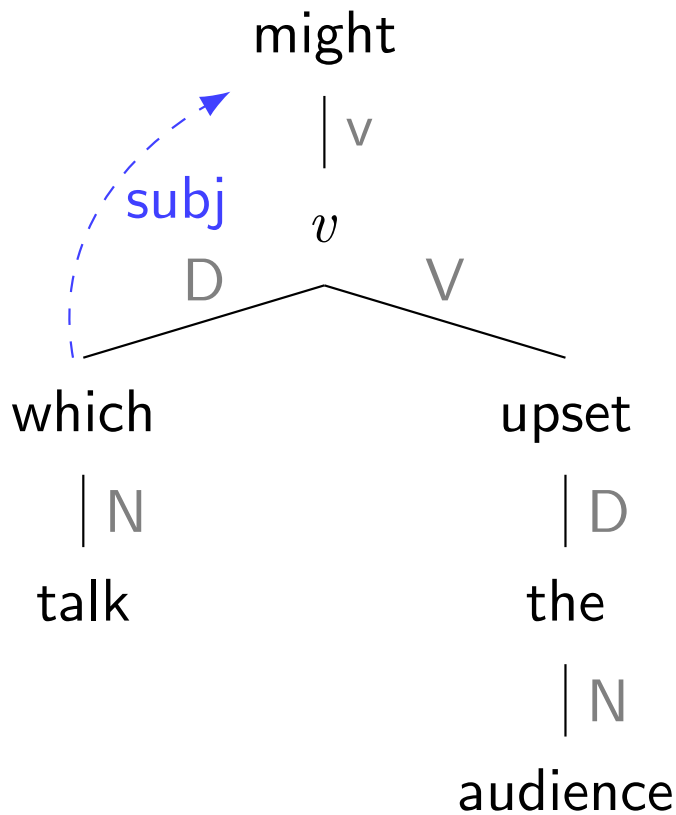




Syntax: Computation, not structures

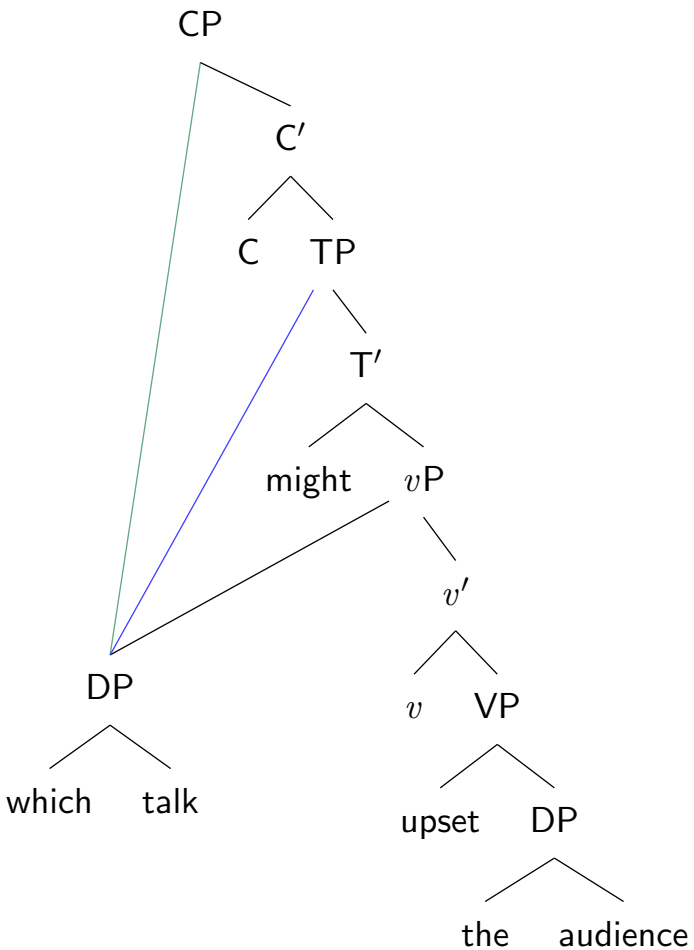
Which talk might upset the audience?

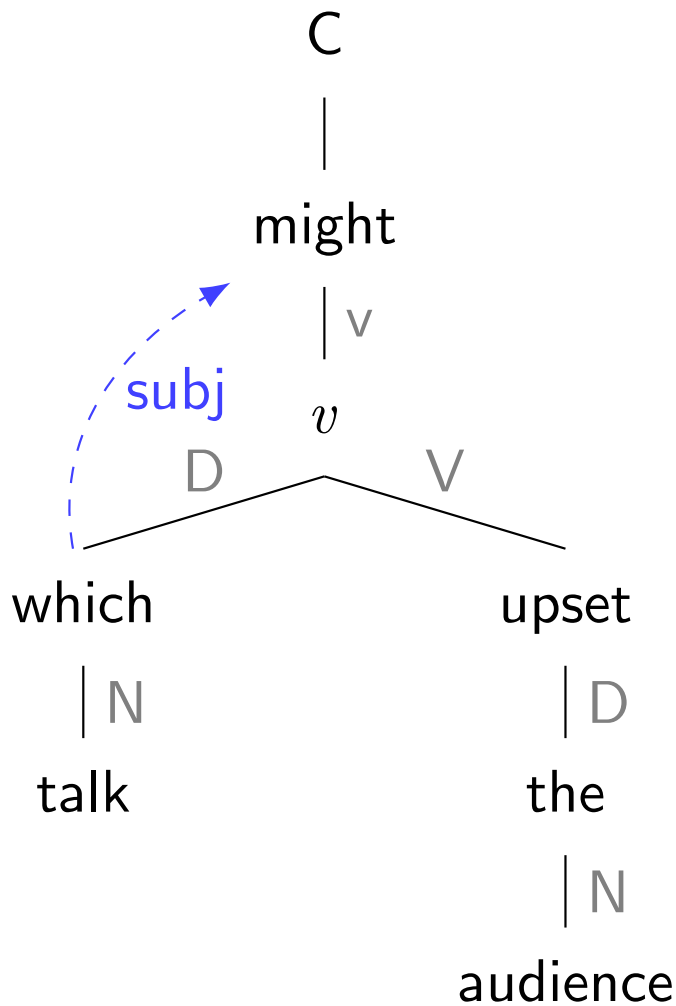




Syntax: Computation, not structures

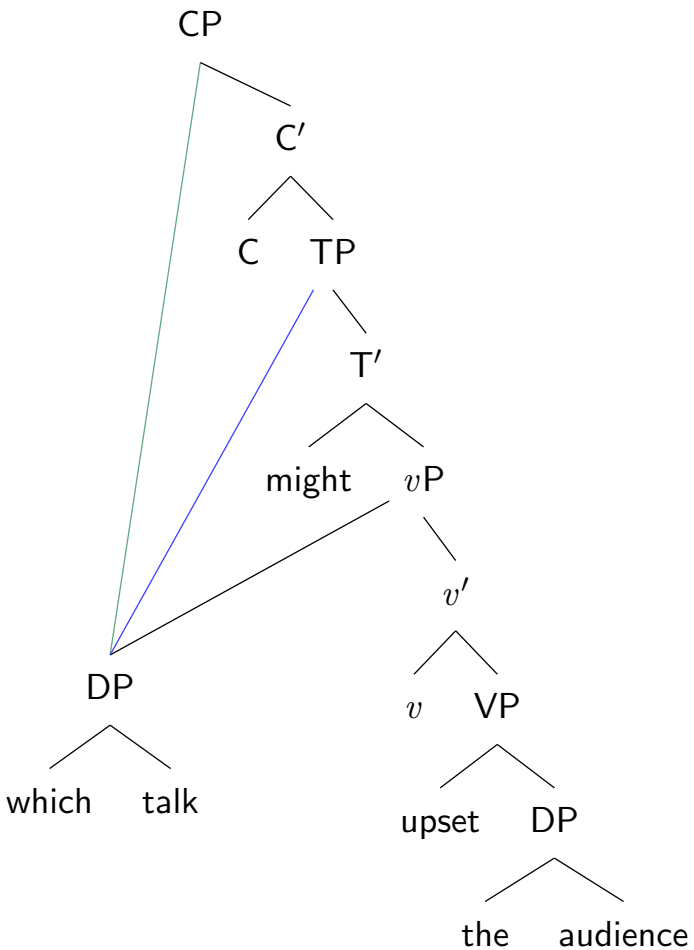
Which talk might upset the audience?

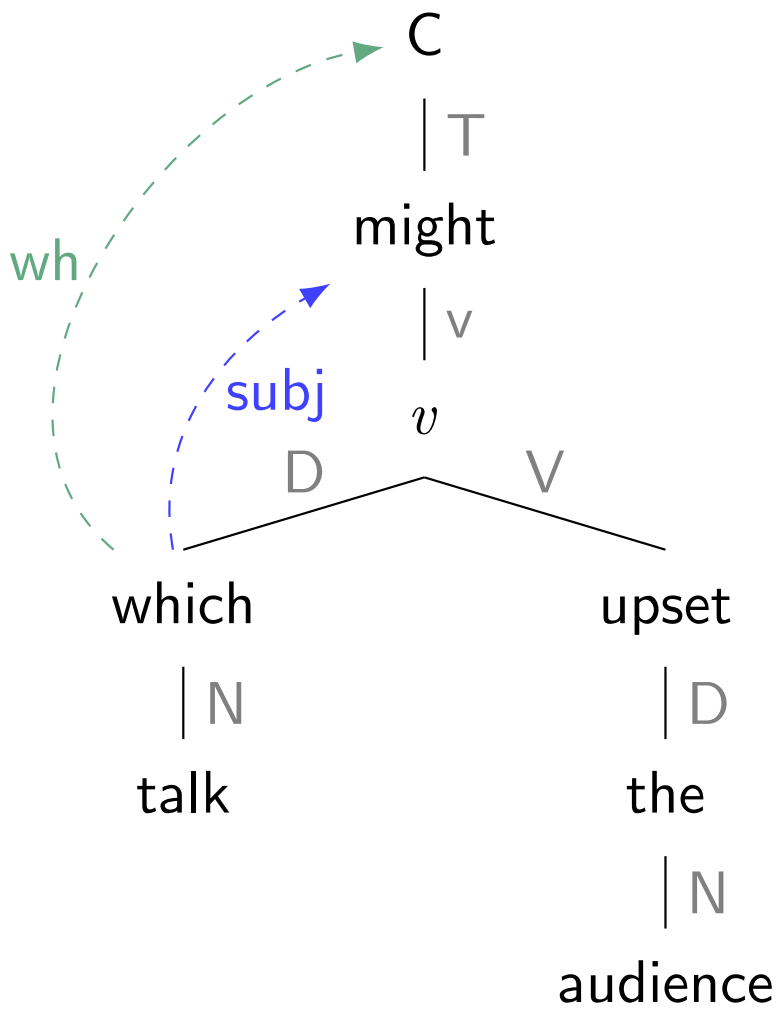




Syntax: Computation, not structures

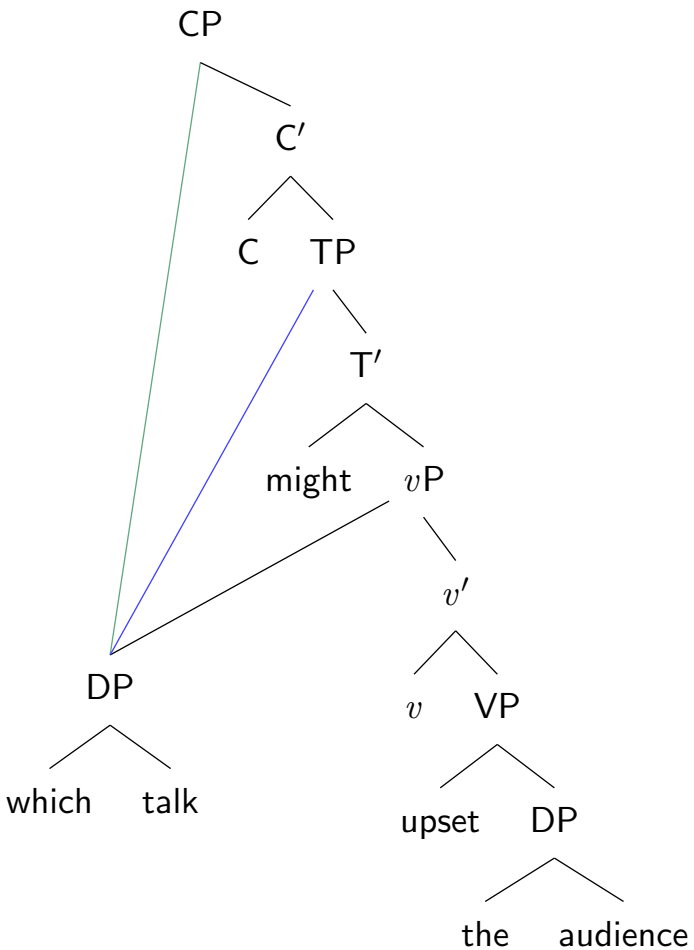
Which talk might upset the audience?

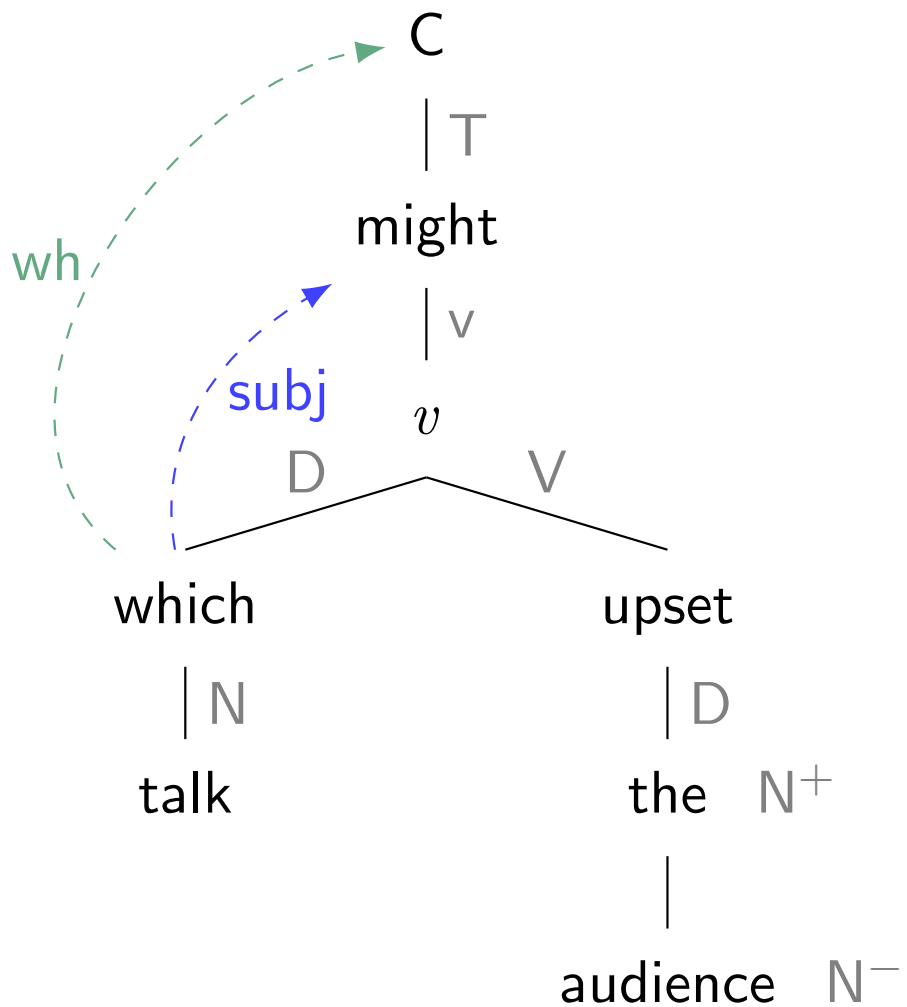




Syntax: Computation, not structures

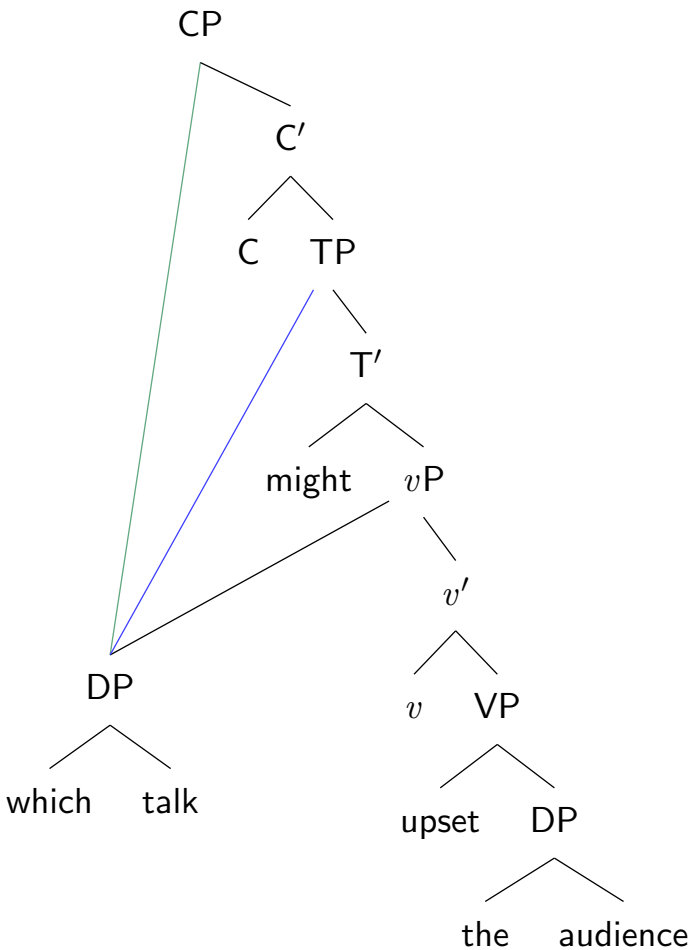
Which talk might upset the audience?

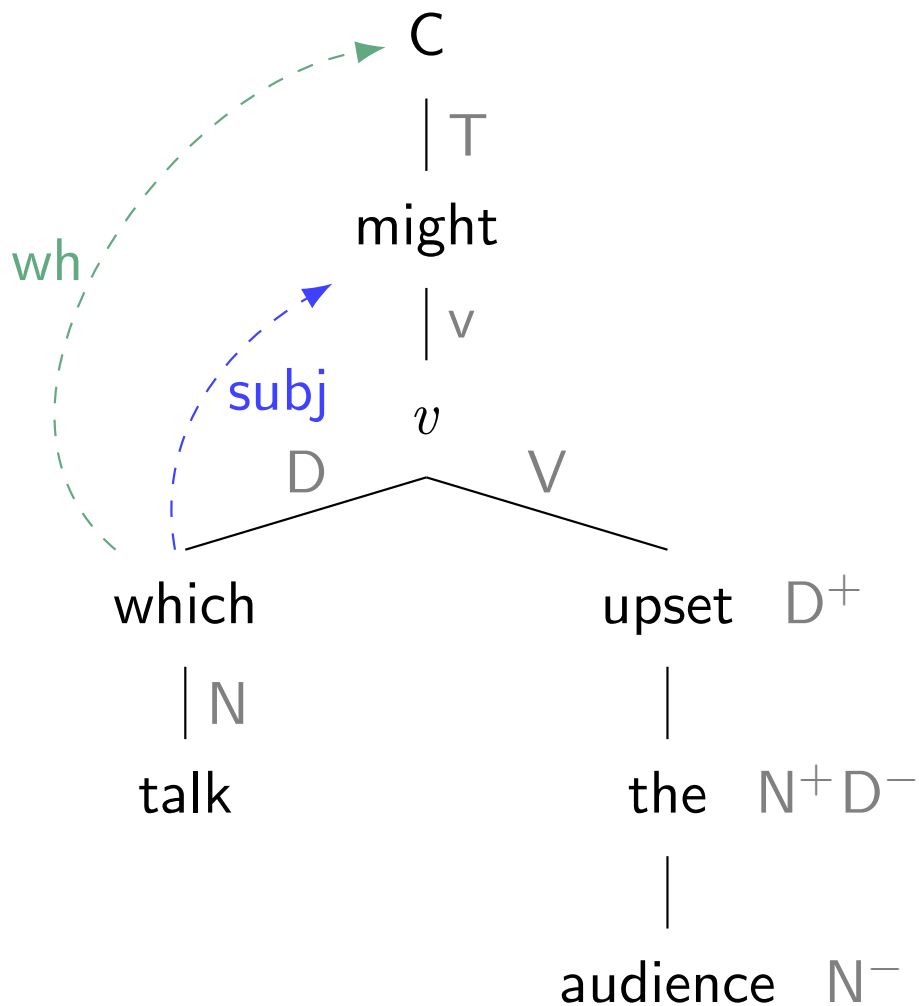




Syntax: Computation, not structures

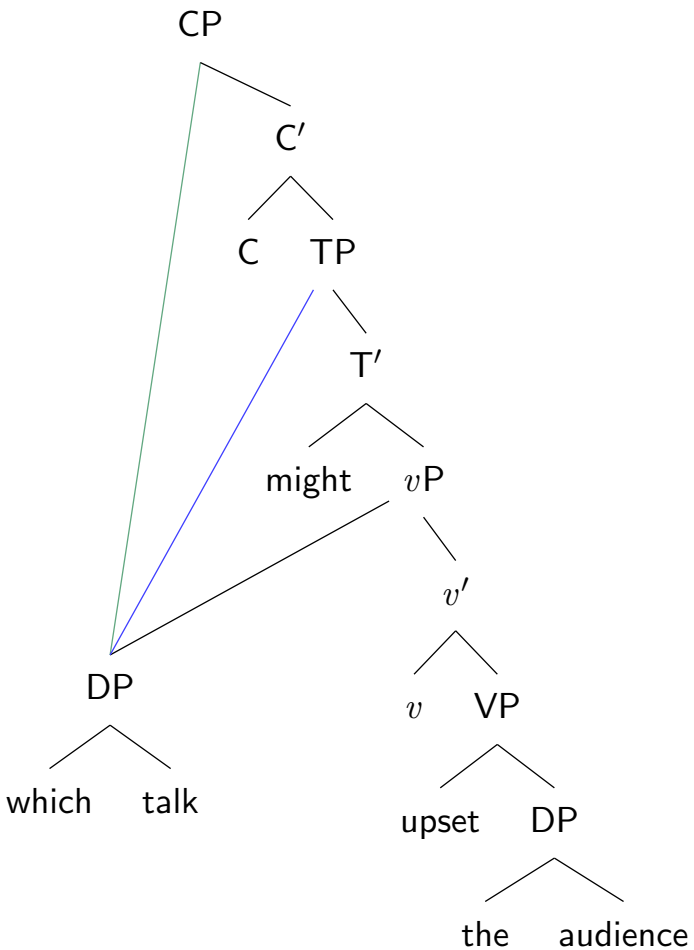
Which talk might upset the audience?

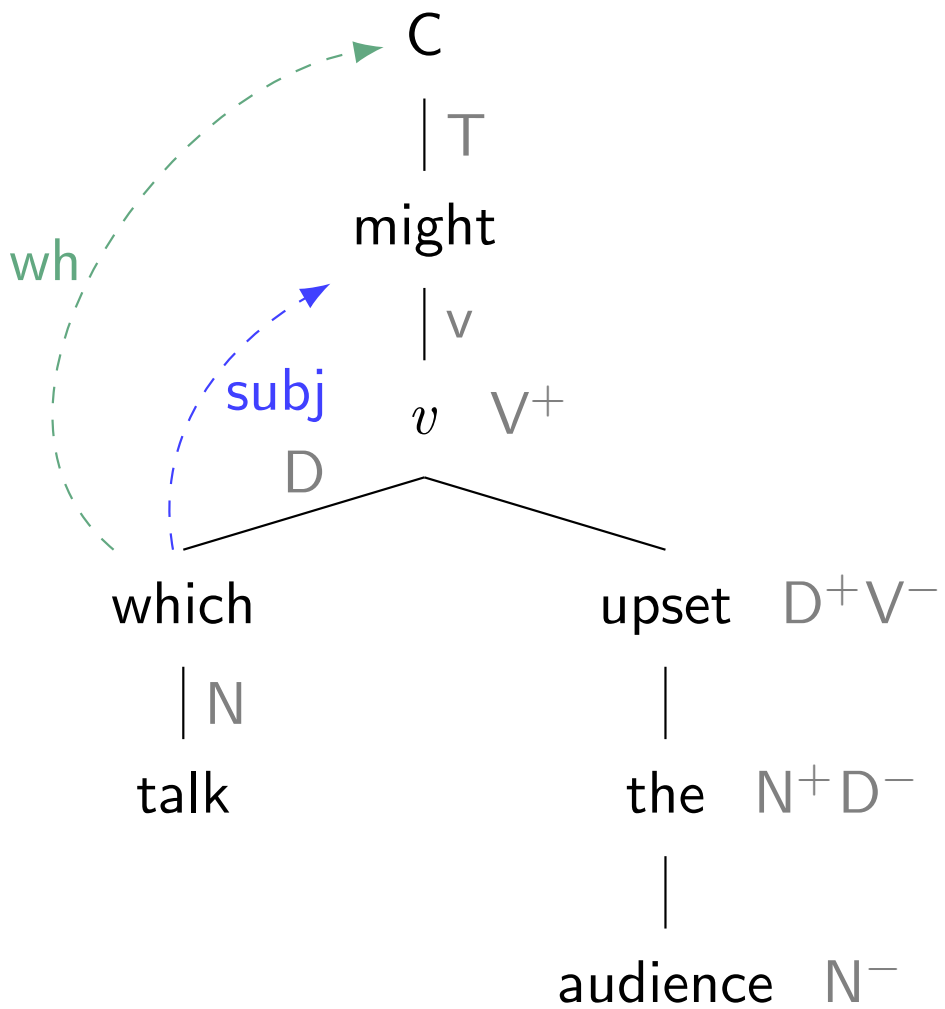




Syntax: Computation, not structures

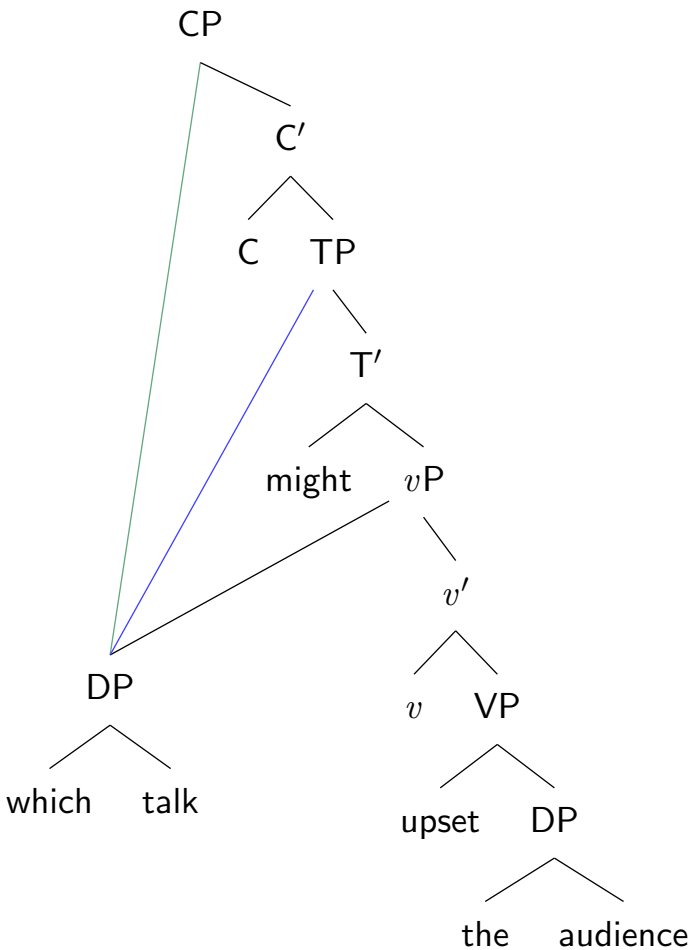
Which talk might upset the audience?

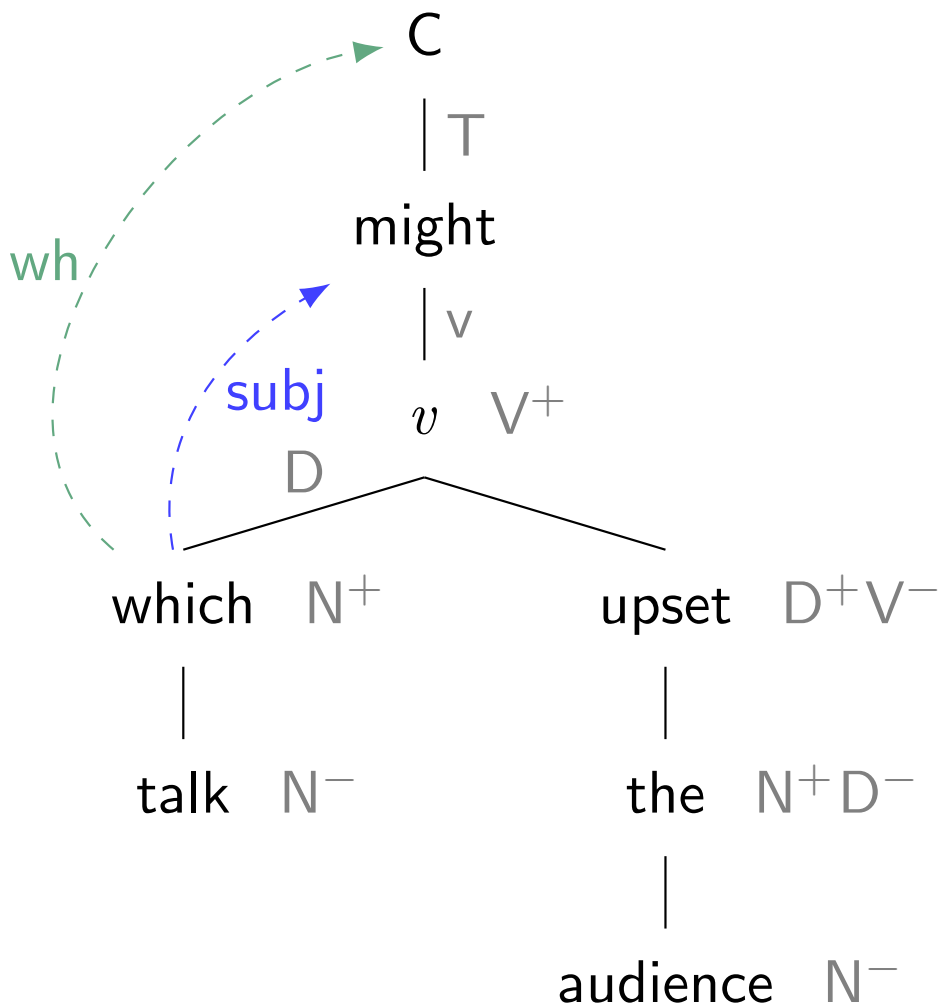




Syntax: Computation, not structures

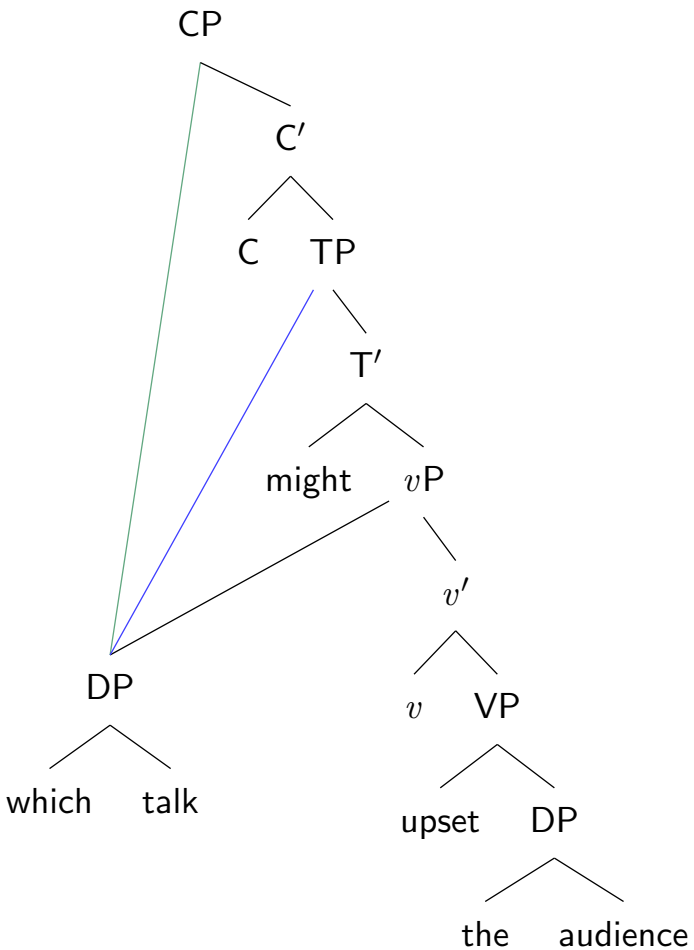
Which talk might upset the audience?

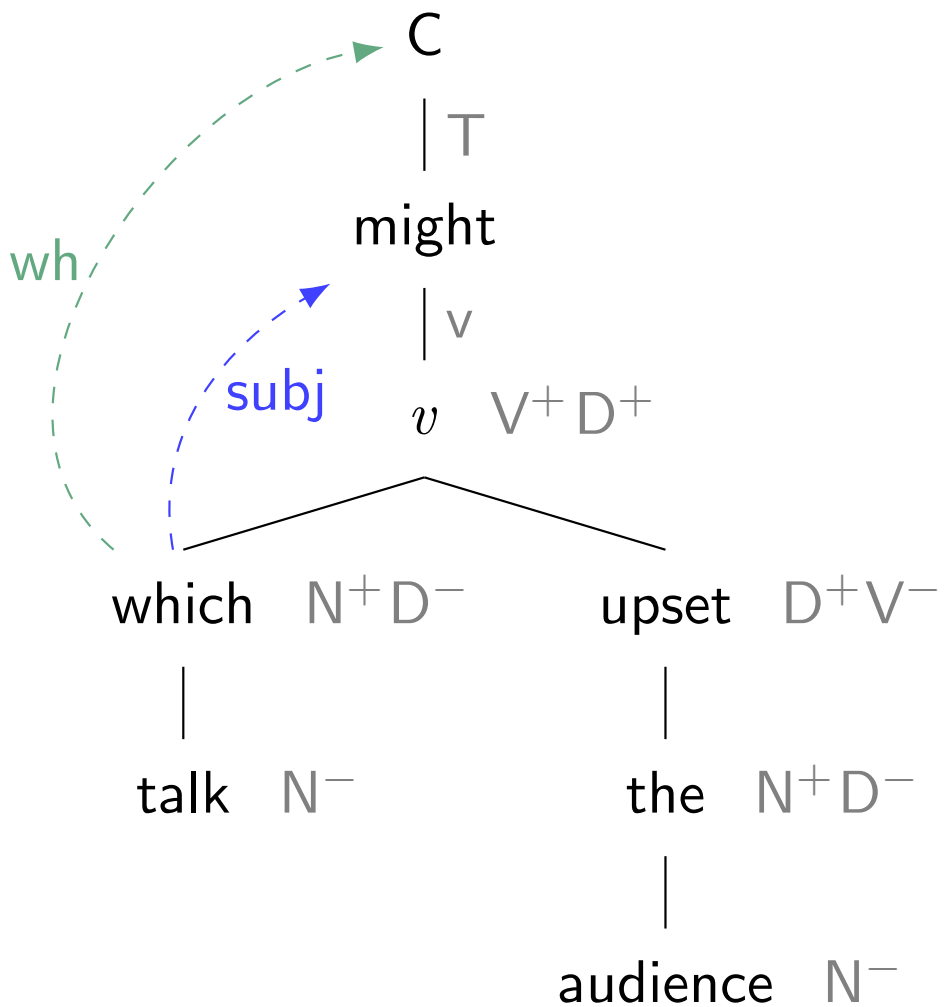




Syntax: Computation, not structures

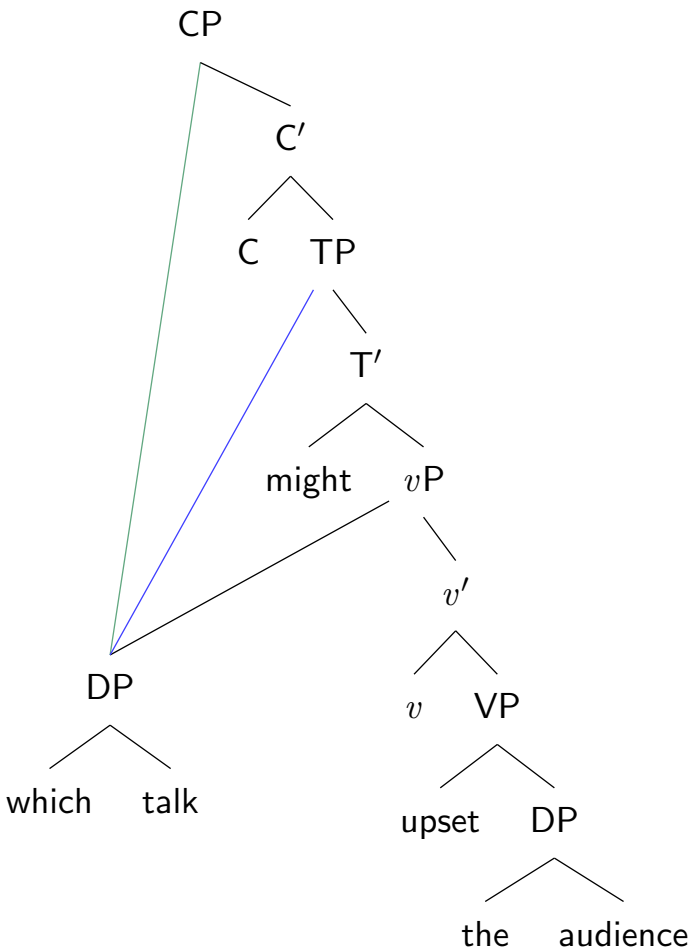
Which talk might upset the audience?

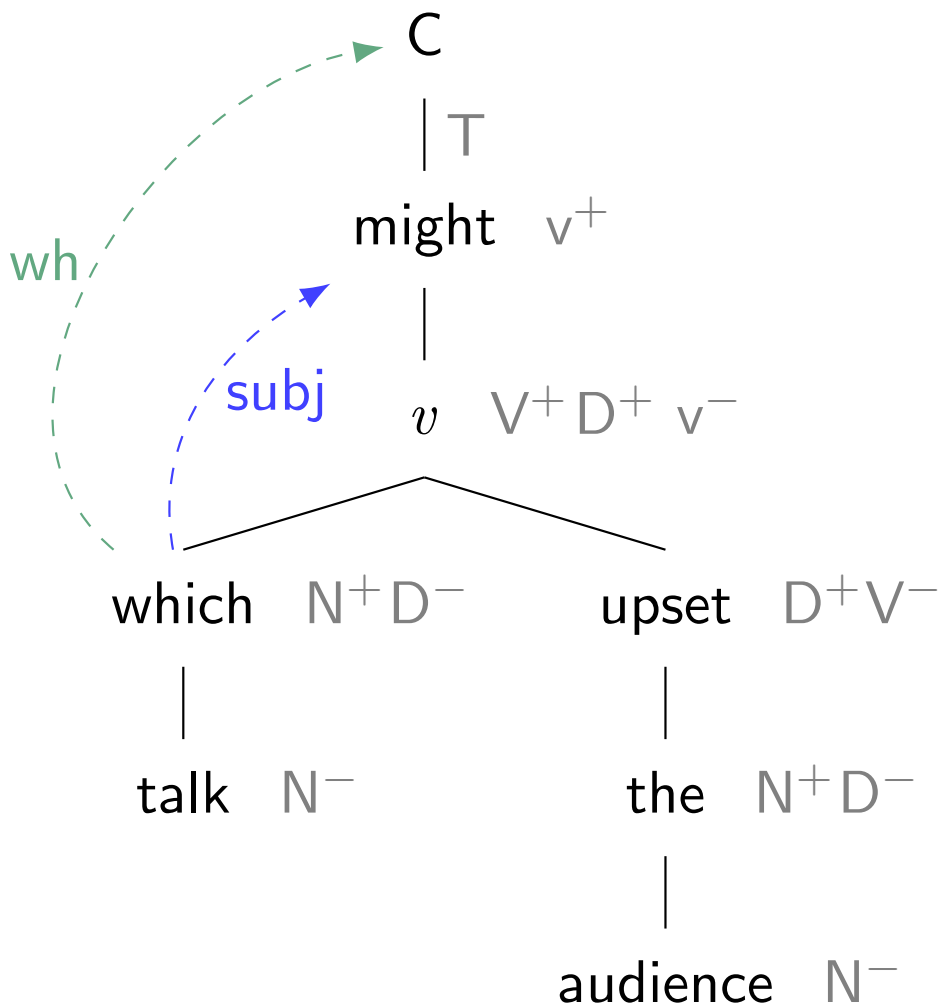




Syntax: Computation, not structures

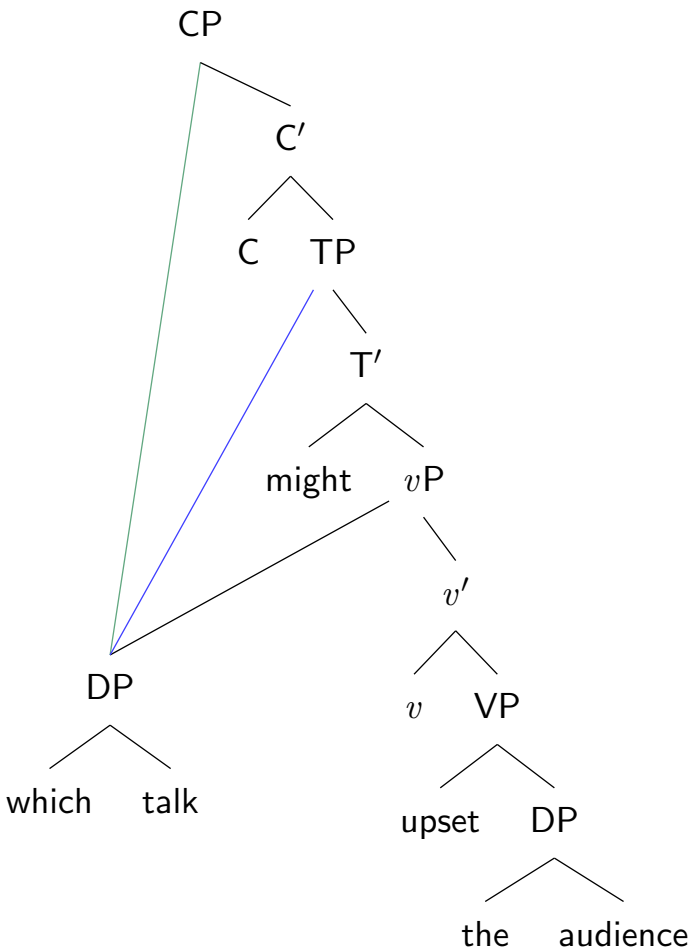
Which talk might upset the audience?

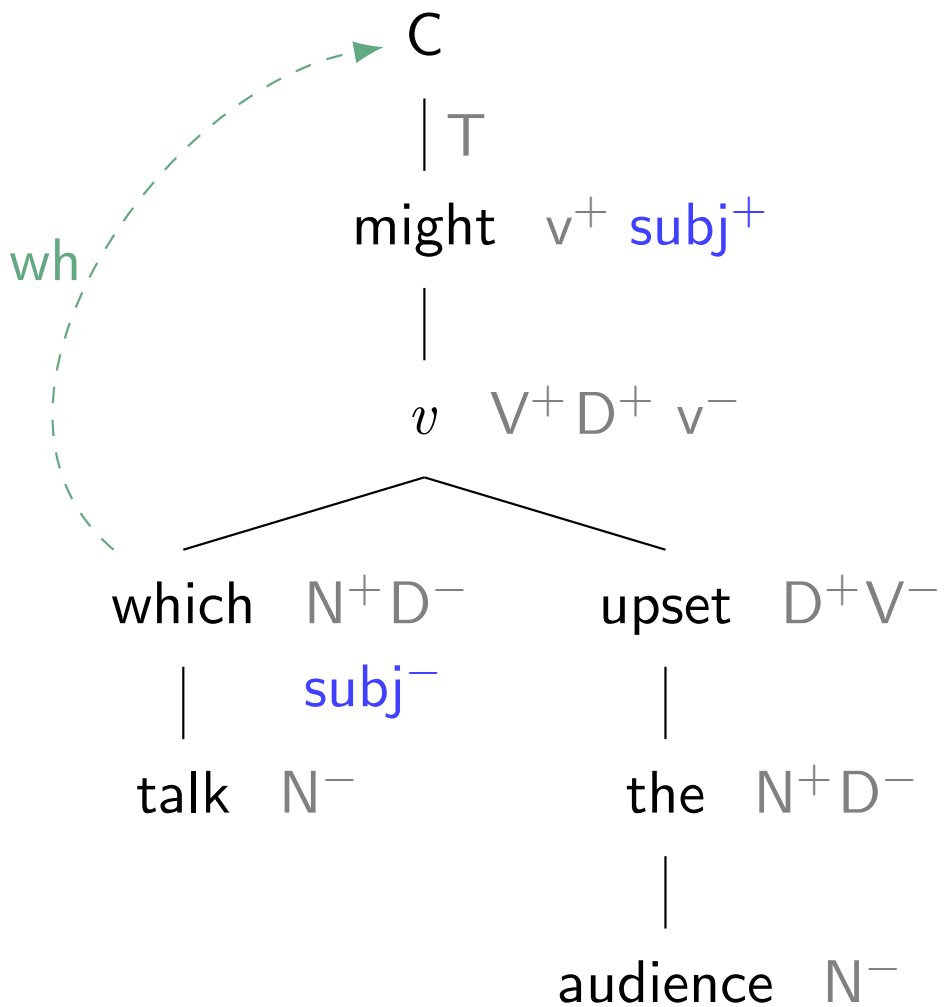




Syntax: Computation, not structures

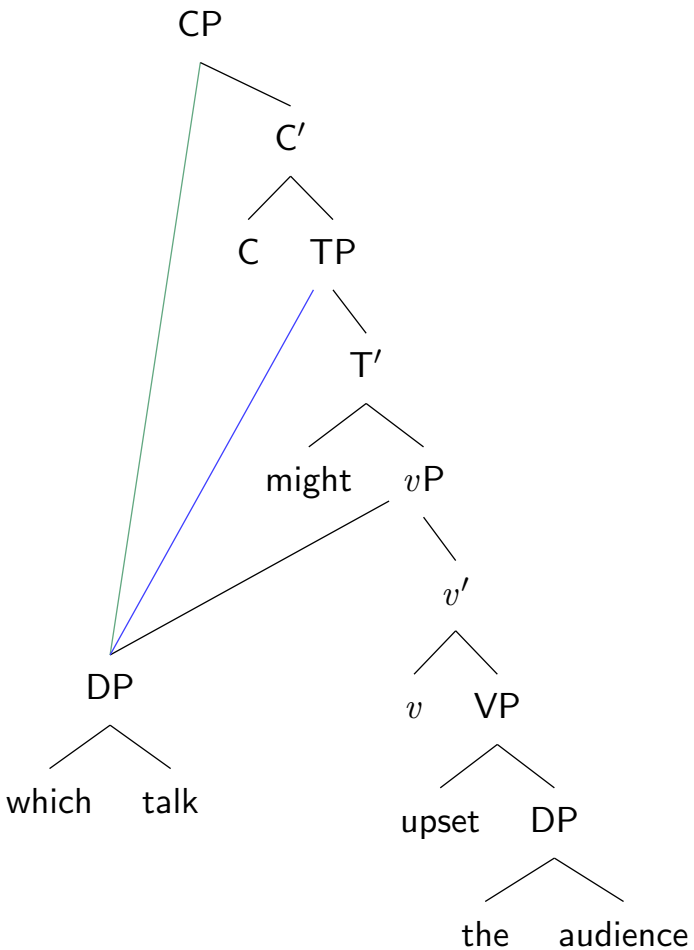
Which talk might upset the audience?

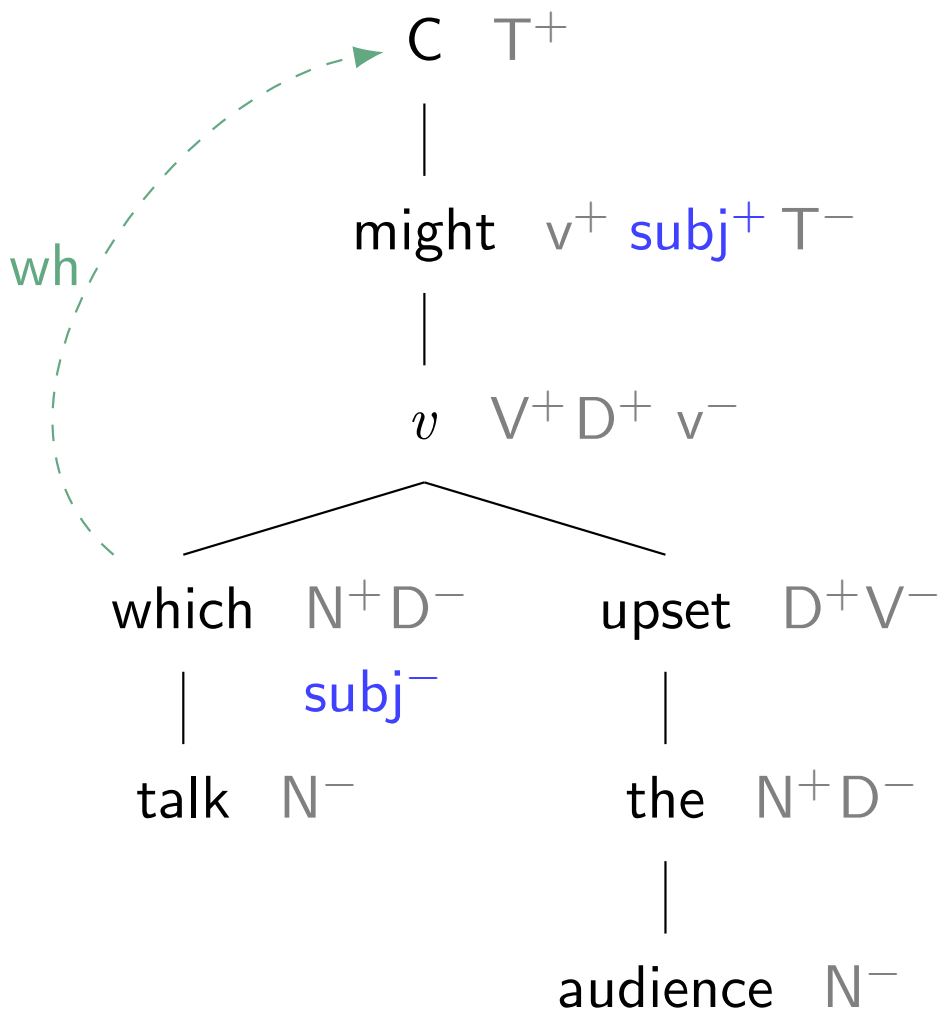




Syntax: Computation, not structures

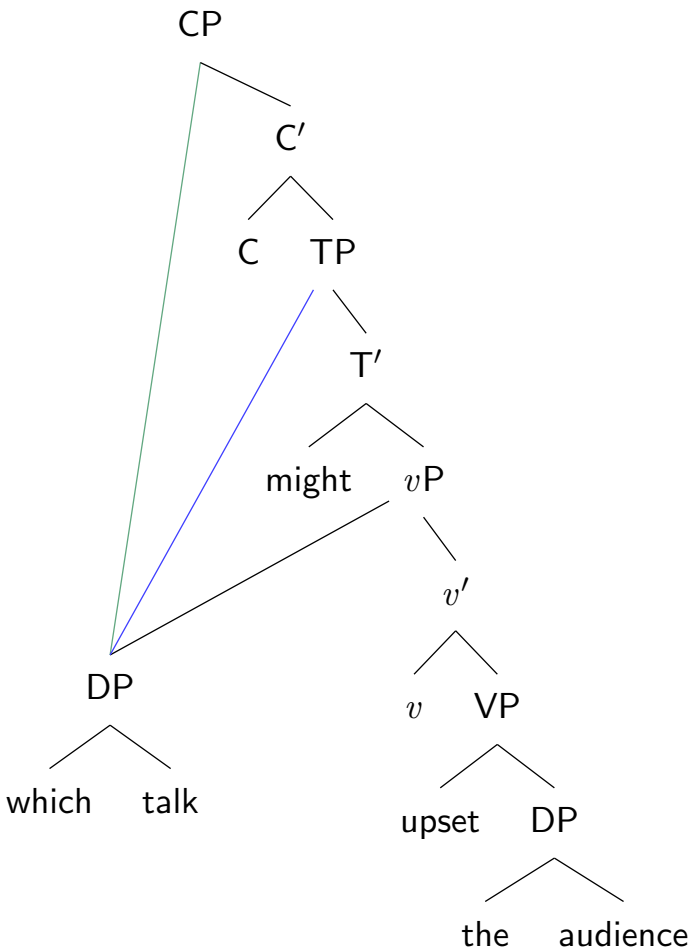
Which talk might upset the audience?

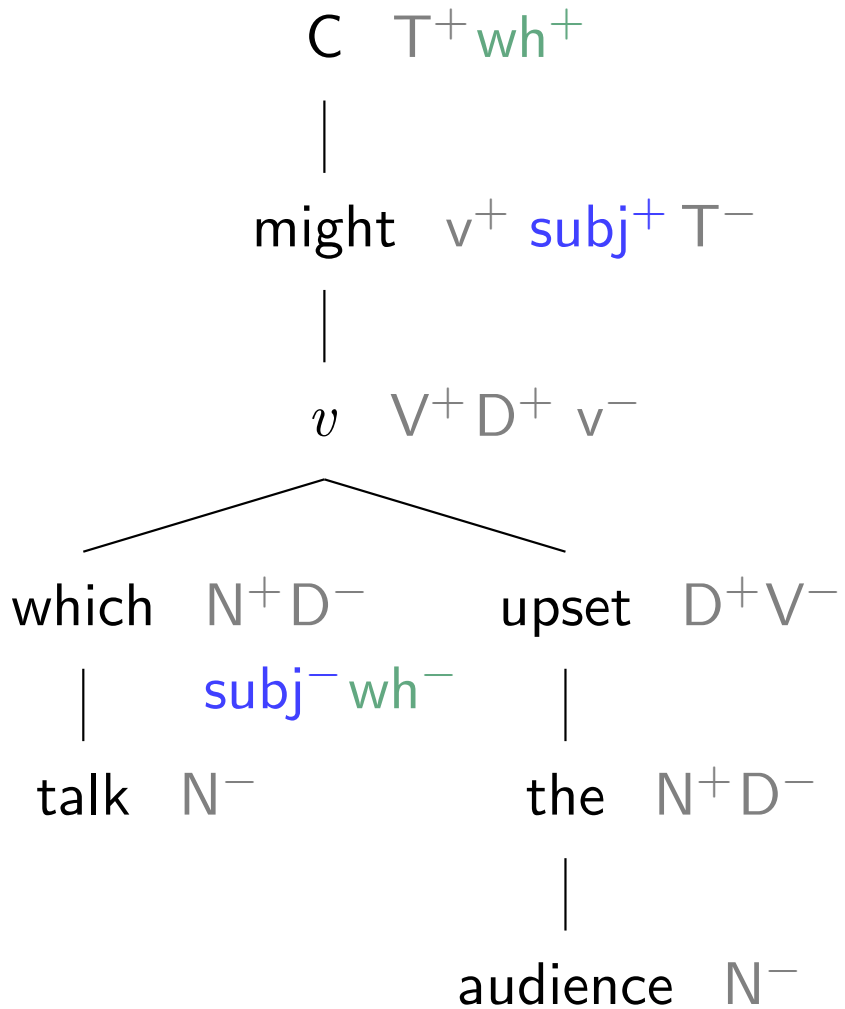




Syntax: Computation, not structures

Which talk might upset the audience?



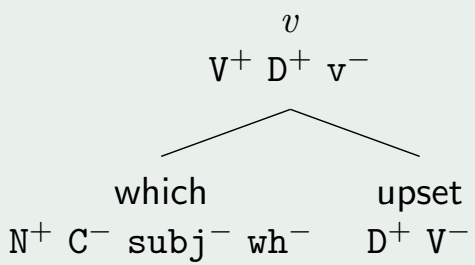


Embrace the abstractness

- "This looks like a dependency tree!"
- Sure does, but it's more abstract: a **computational trace**

Different interpretations of the same computational trace

- v has the dependents *which* and *upset*
- Merge(v, *upset*), then Merge(v, *which*)
- 3-Merge(v, *upset*, *which*)
- Split v into *which* and *upset*



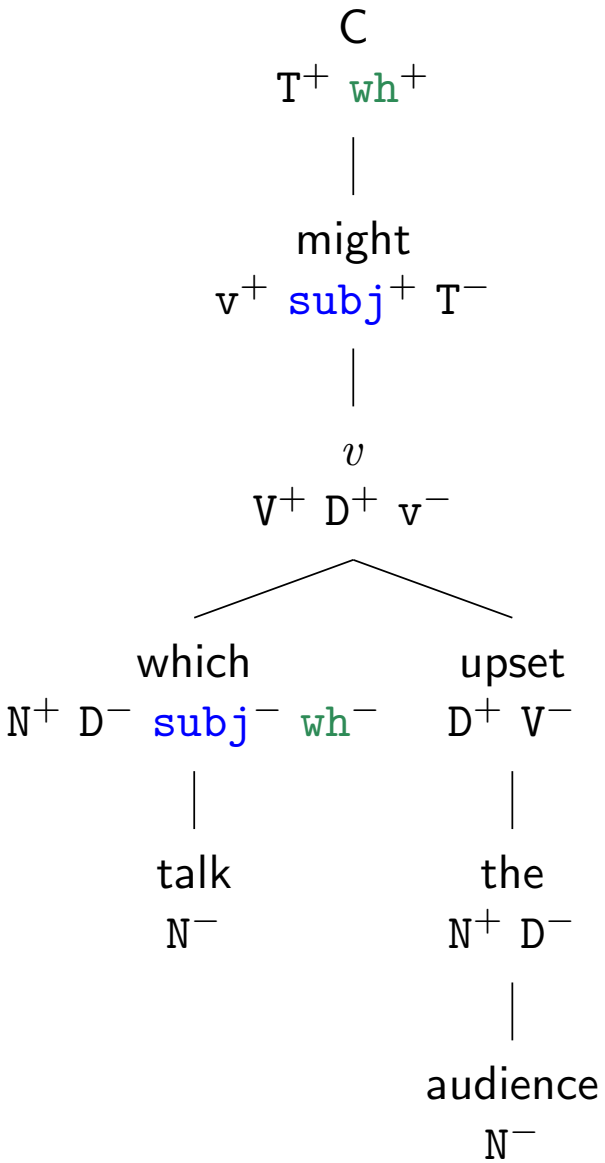
- The trace records the computational common core of all these different systems.
- What can we say about this common core?

Establishing a computational baseline

- **The Ceiling Constraint**
  Every $\texttt{f}^-$ needs an $\texttt{f}^+$
- **A different formulation**
  For every node **n** and every $\texttt{f}^-$:
  if **n** carries $\texttt{f}^-$, then its **a-string**
  contains a node with $\texttt{f}^+$.

Definition (a[ncestor]-string; simplified)

The **a-string** of **n** consists of all the nodes that are derivationally ordered after **n**.

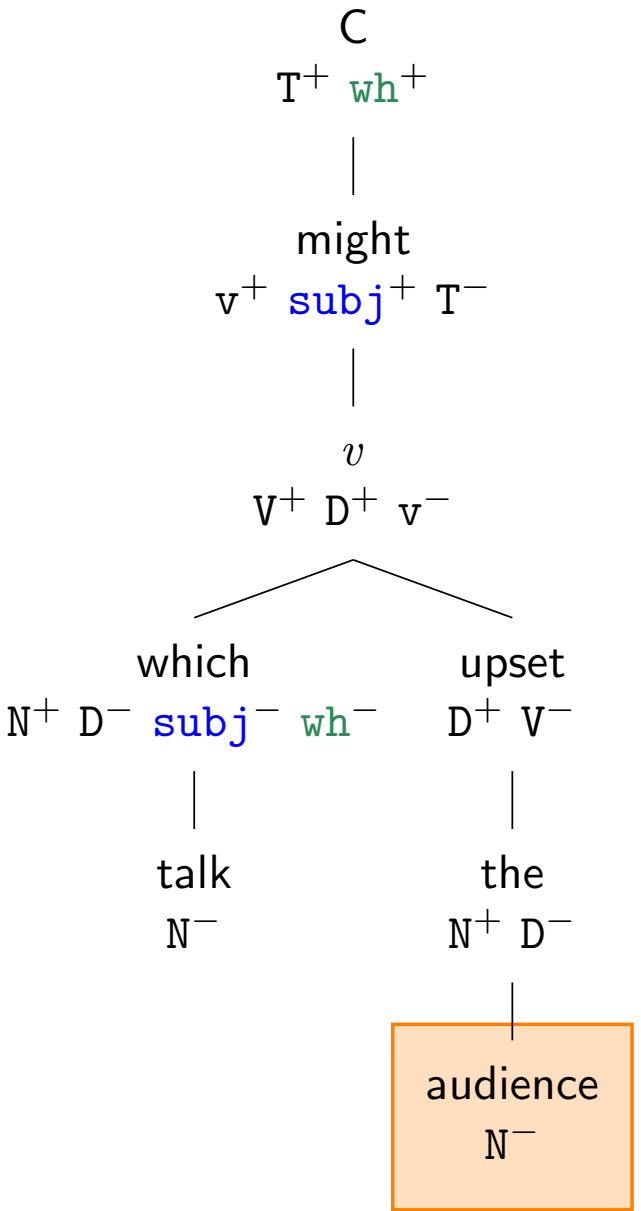


Establishing a computational baseline

- **The Ceiling Constraint**
  Every $\mathtt{f}^-$ needs an $\mathtt{f}^+$
- **A different formulation**
  For every node **n** and every $\mathtt{f}^-$:
  if **n** carries $\mathtt{f}^-$, then its **a-string**
  contains a node with $\mathtt{f}^+$.

Definition (a[ncestor]-string; simplified)

The **a-string** of **n** consists of all the nodes that are derivationally ordered after **n**.

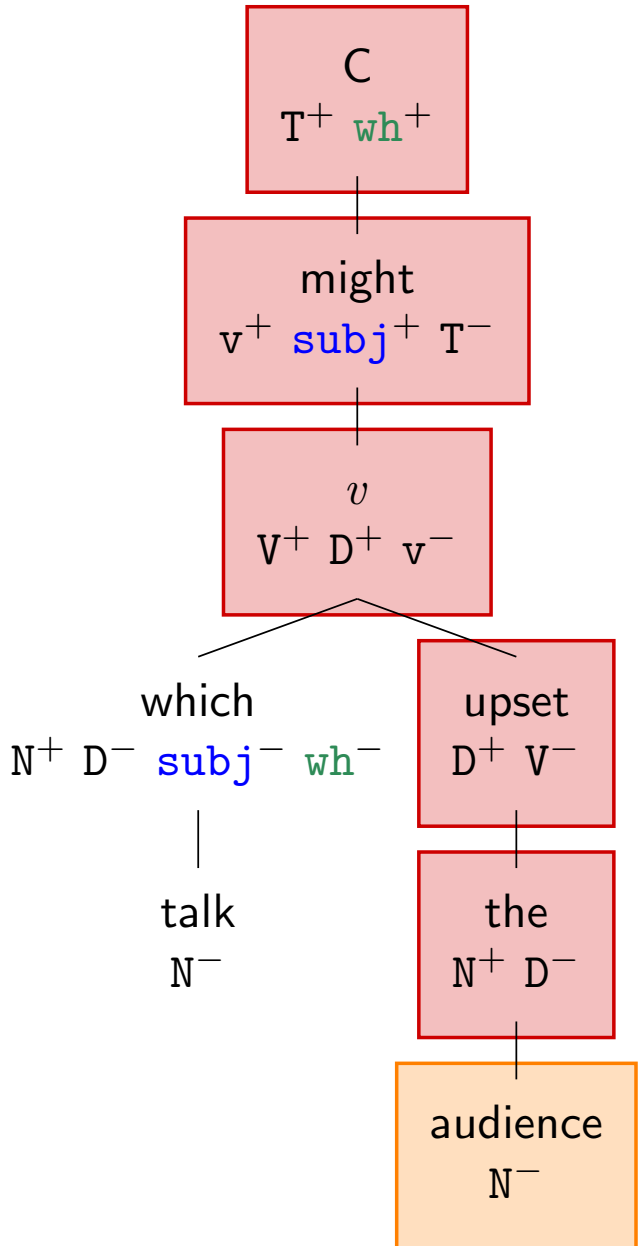


Establishing a computational baseline

- **The Ceiling Constraint**
  Every $\mathtt{f}^-$ needs an $\mathtt{f}^+$
- **A different formulation**
  For every node **n** and every $\mathtt{f}^-$:
  if **n** carries $\mathtt{f}^-$, then its **a-string**
  contains a node with $\mathtt{f}^+$.

Definition (a[ncestor]-string; simplified)

The **a-string** of **n** consists of all the nodes that are derivationally ordered after **n**.

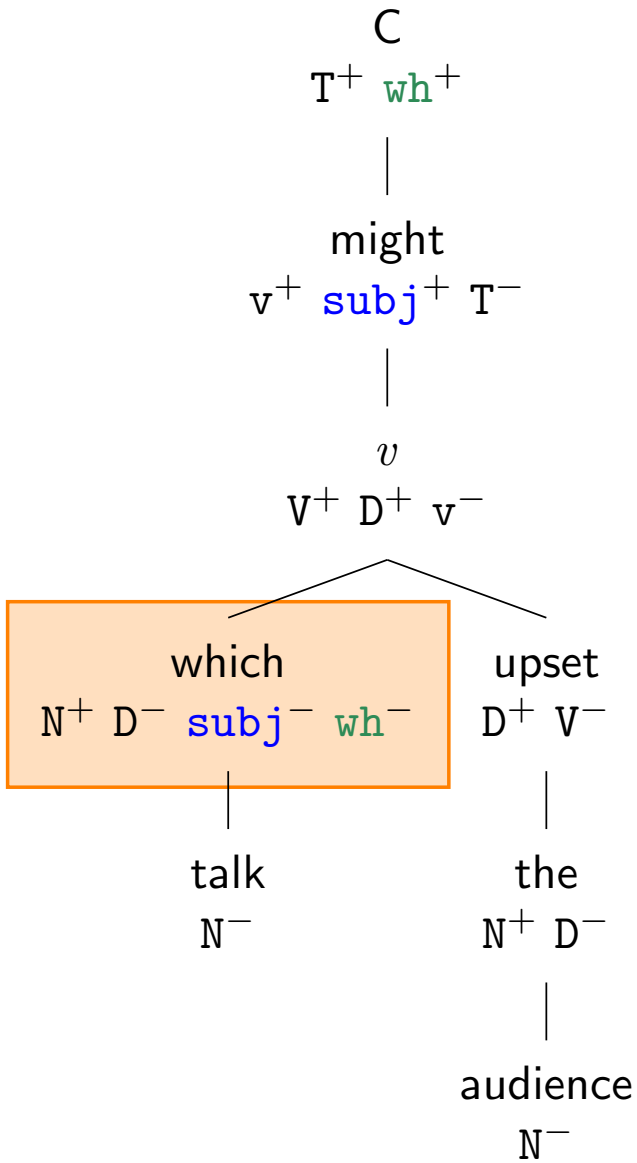


Establishing a computational baseline

- **The Ceiling Constraint**
  Every $\texttt{f}^-$ needs an $\texttt{f}^+$
- **A different formulation**
  For every node **n** and every $\texttt{f}^-$:
  if **n** carries $\texttt{f}^-$, then its **a-string**
  contains a node with $\texttt{f}^+$.

Definition (a[ncestor]-string; simplified)

The **a-string** of **n** consists of all the nodes that are derivationally ordered after **n**.

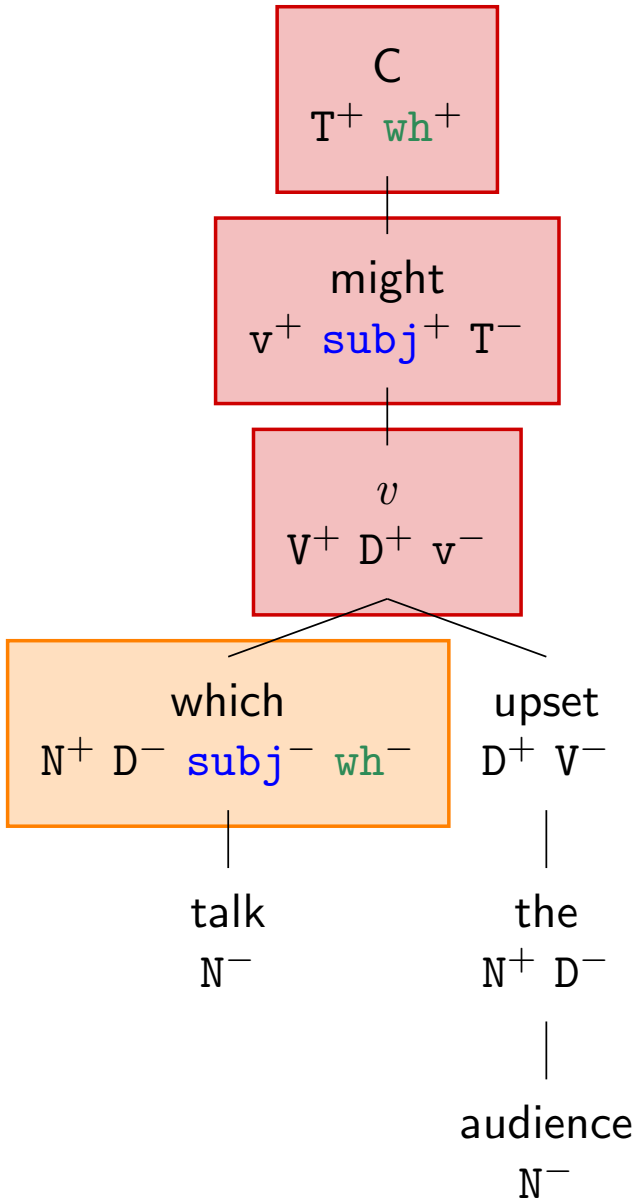


Establishing a computational baseline

- **The Ceiling Constraint**
  Every $\mathtt{f}^-$ needs an $\mathtt{f}^+$
- **A different formulation**
  For every node **n** and every $\mathtt{f}^-$:
  if **n** carries $\mathtt{f}^-$, then its **a-string**
  contains a node with $\mathtt{f}^+$.

Definition (a[ncestor]-string; simplified)

The **a-string** of **n** consists of all the nodes that are derivationally ordered after **n**.



Telling whether a string contains $\mathtt{f}^+$

- The a-string of a node with $\mathtt{f}^-$ must belong to the string set $\Sigma^*\mathtt{f}^+\Sigma^*$.
  - Σ^*: any arbitrary string of 0 or more nodes
  - $\mathtt{f}^+$: any one node that carries $\mathtt{f}^+$
- This is one of the simplest possible string sets, it is **tier-based reverse 1-definite**:
  1. Project an $\mathtt{f}$-tier
  2. The first symbol must be $\mathtt{f}^+$

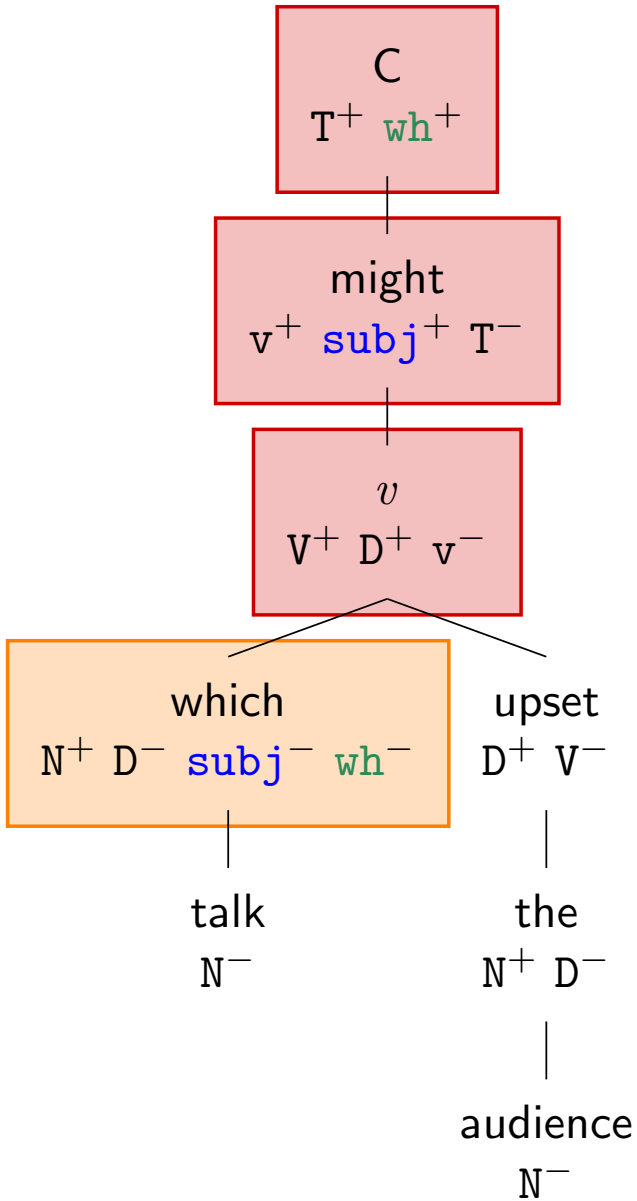
(Lambert 2022)

Dakotah Lambert

A small example

1. **Checking $\texttt{subj}^-$ on which**

2. **Checking $\texttt{wh}^-$ on which**

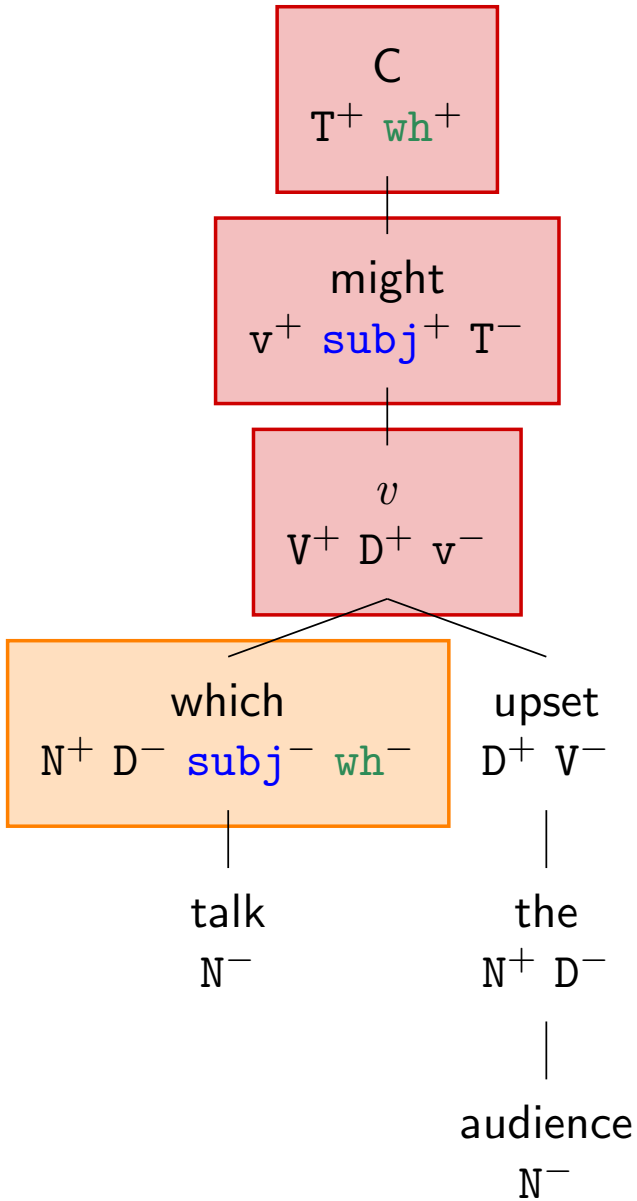


A small example

1. **Checking $\texttt{subj}^-$ on which**

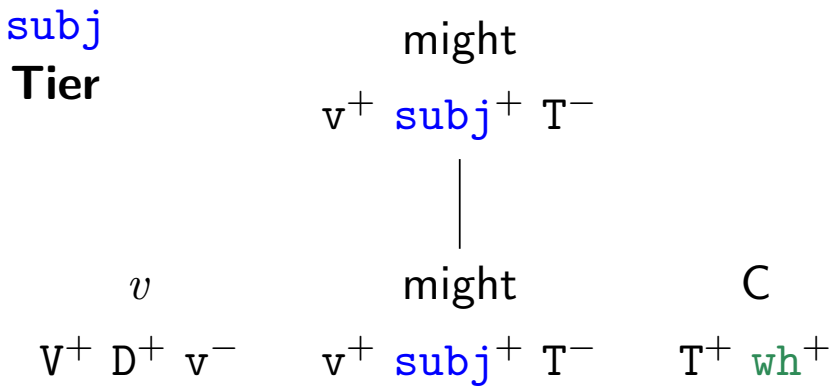
v	might	C
$\texttt{V}^+\ \texttt{D}^+\ \texttt{v}^-$	$\texttt{v}^+\ \texttt{subj}^+\ \texttt{T}^-$	$\texttt{T}^+\ \texttt{wh}^+$

2. **Checking $\texttt{wh}^-$ on which**

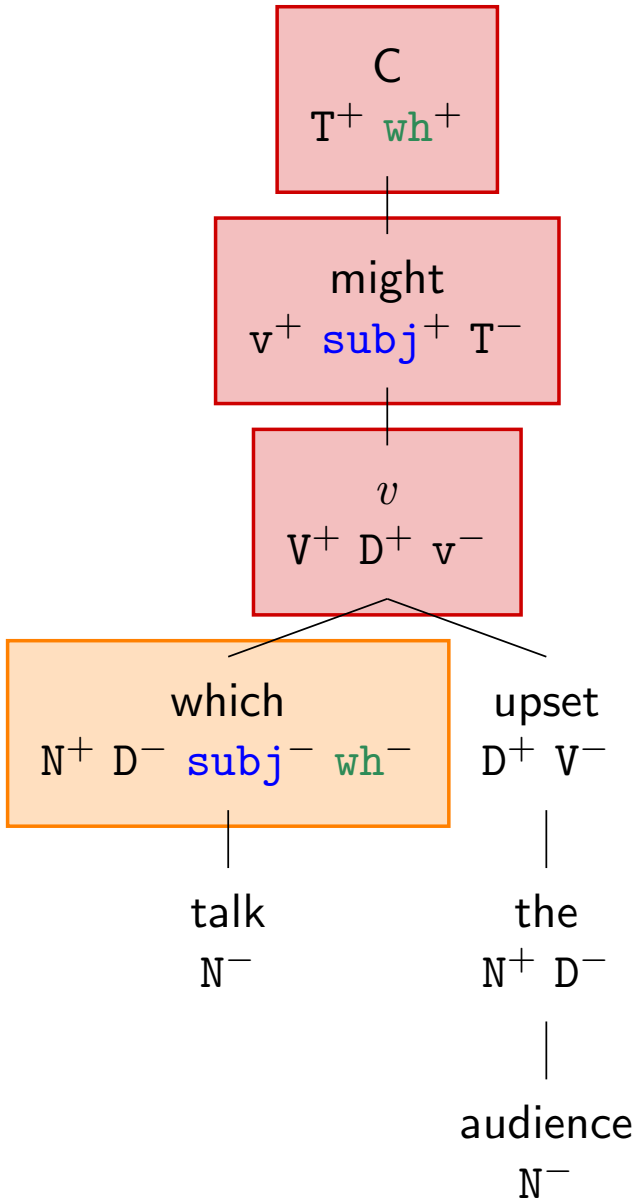


A small example

1. **Checking $\texttt{subj}^-$ on which**



2. **Checking $\texttt{wh}^-$ on which**



A small example

1 Checking $\texttt{subj}^-$ on which

subj Tier

might
$\texttt{v}^+\ \texttt{subj}^+\ \texttt{T}^-$

v
$\texttt{V}^+\ \texttt{D}^+\ \texttt{v}^-$

might
$\texttt{v}^+\ \texttt{subj}^+\ \texttt{T}^-$

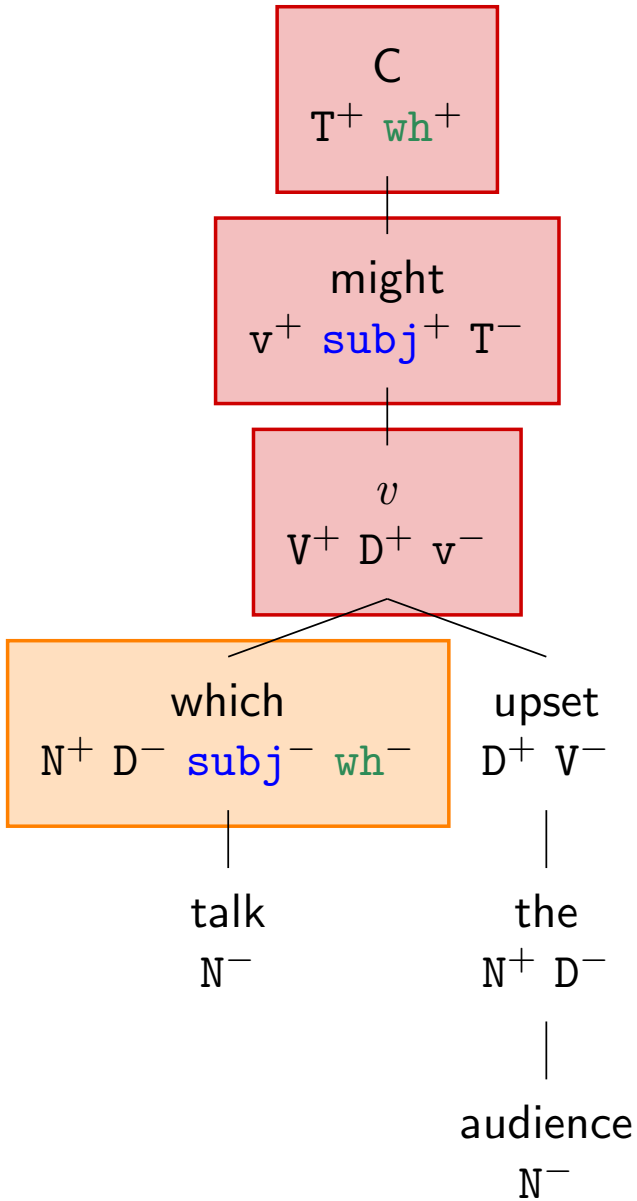
C
$\texttt{T}^+\ \texttt{wh}^+$

2 Checking $\texttt{wh}^-$ on which

v
$\texttt{V}^+\ \texttt{D}^+\ \texttt{v}^-$

might
$\texttt{v}^+\ \texttt{subj}^+\ \texttt{T}^-$

C
$\texttt{T}^+\ \texttt{wh}^+$



A small example

1 **Checking $\texttt{subj}^-$ on which**

subj
Tier

might
$\texttt{v}^+\ \texttt{subj}^+\ \texttt{T}^-$

v
$\texttt{V}^+\ \texttt{D}^+\ \texttt{v}^-$

might
$\texttt{v}^+\ \texttt{subj}^+\ \texttt{T}^-$

C
$\texttt{T}^+\ \texttt{wh}^+$

2 **Checking $\texttt{wh}^-$ on which**

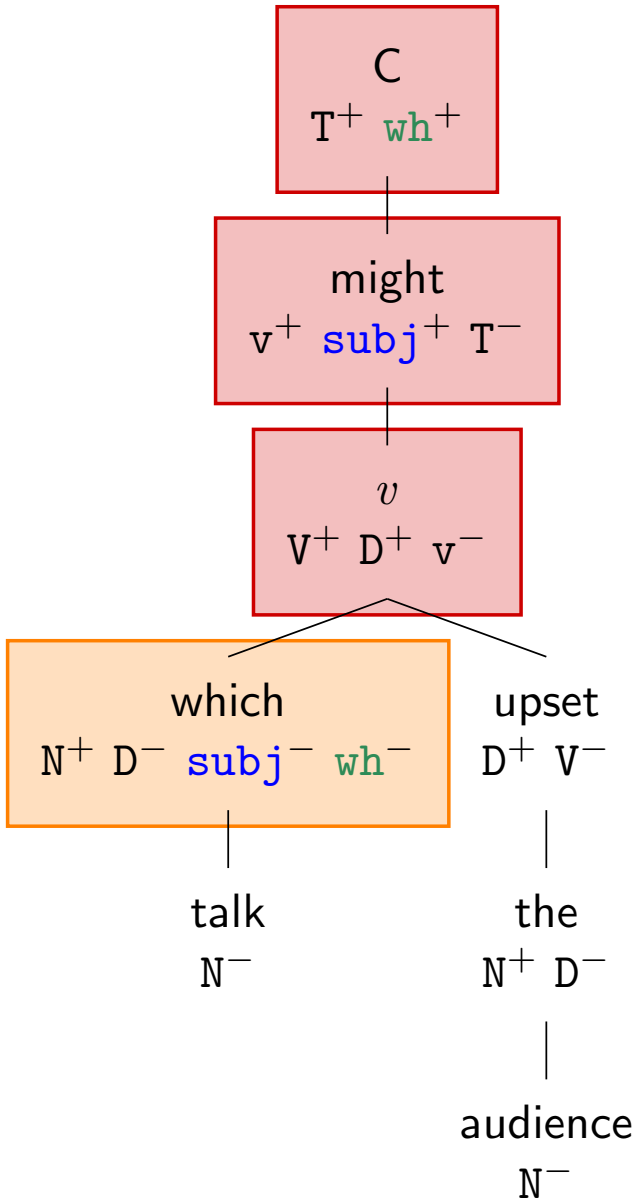
wh
Tier

C
$\texttt{T}^+\ \texttt{wh}^+$

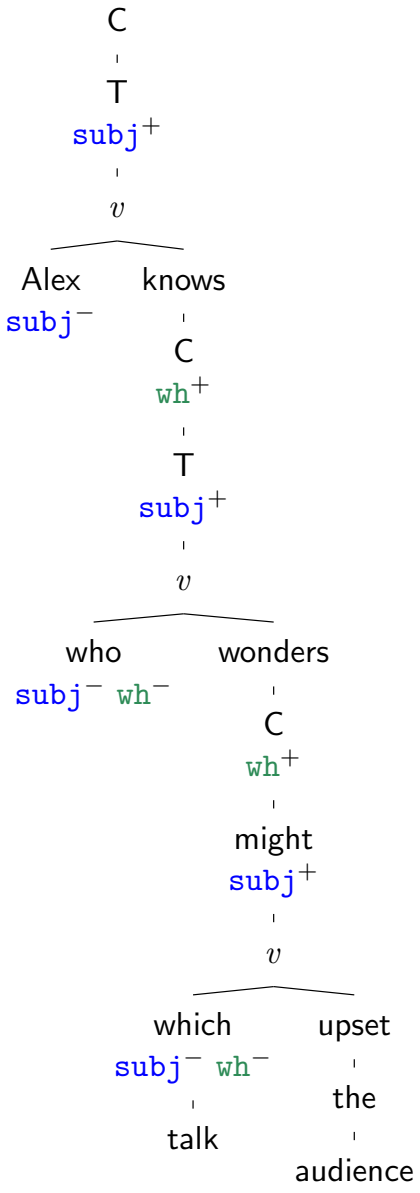
v
$\texttt{V}^+\ \texttt{D}^+\ \texttt{v}^-$

might
$\texttt{v}^+\ \texttt{subj}^+\ \texttt{T}^-$

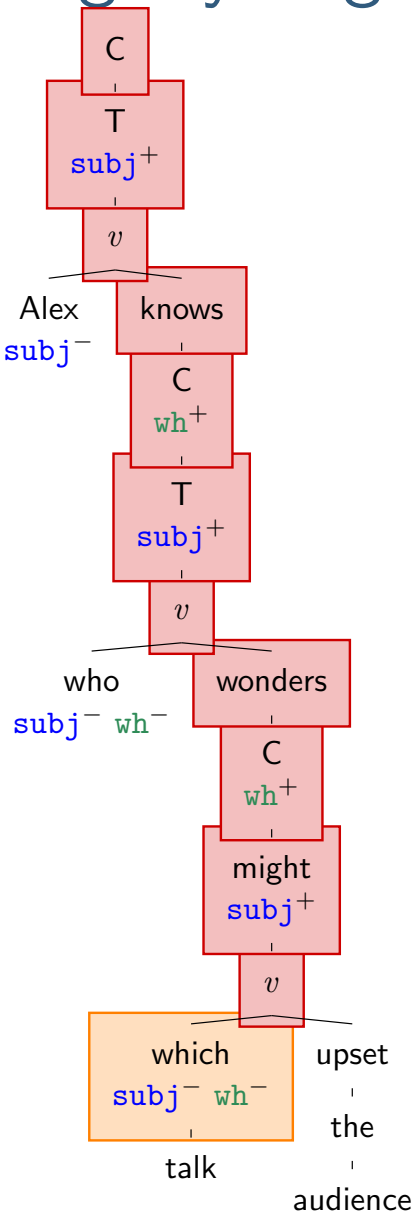
C
$\texttt{T}^+\ \texttt{wh}^+$



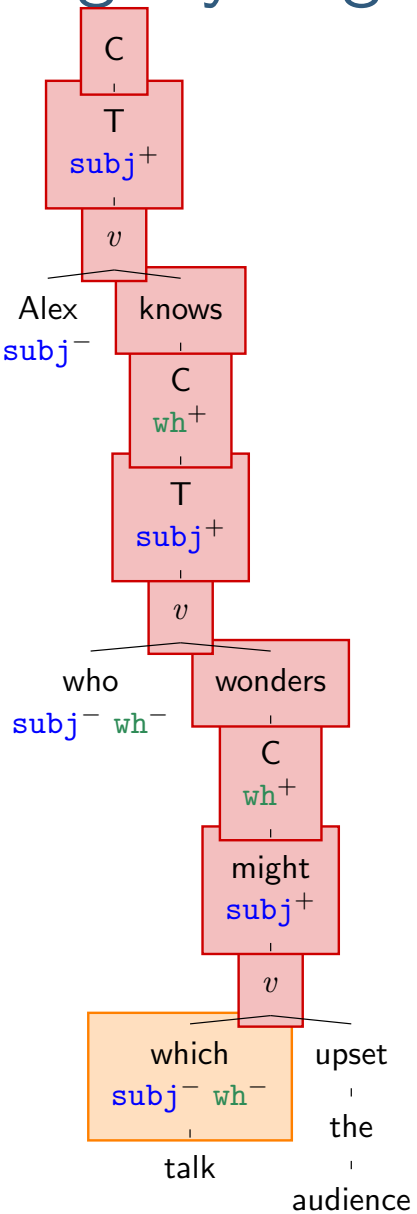
A slightly larger example



A slightly larger example



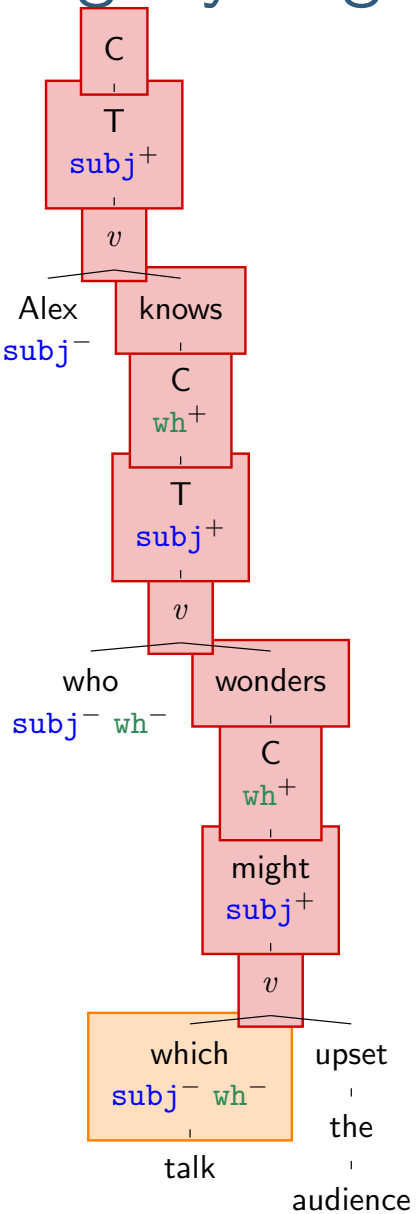
A slightly larger example



v might C wonders v T C knows v T C

subj+ wh+ subj+ wh+ subj+

A slightly larger example

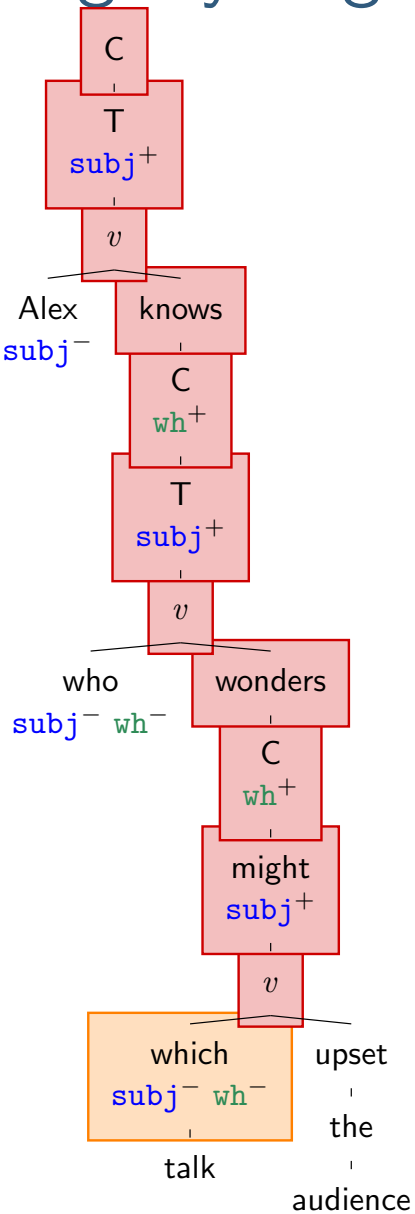


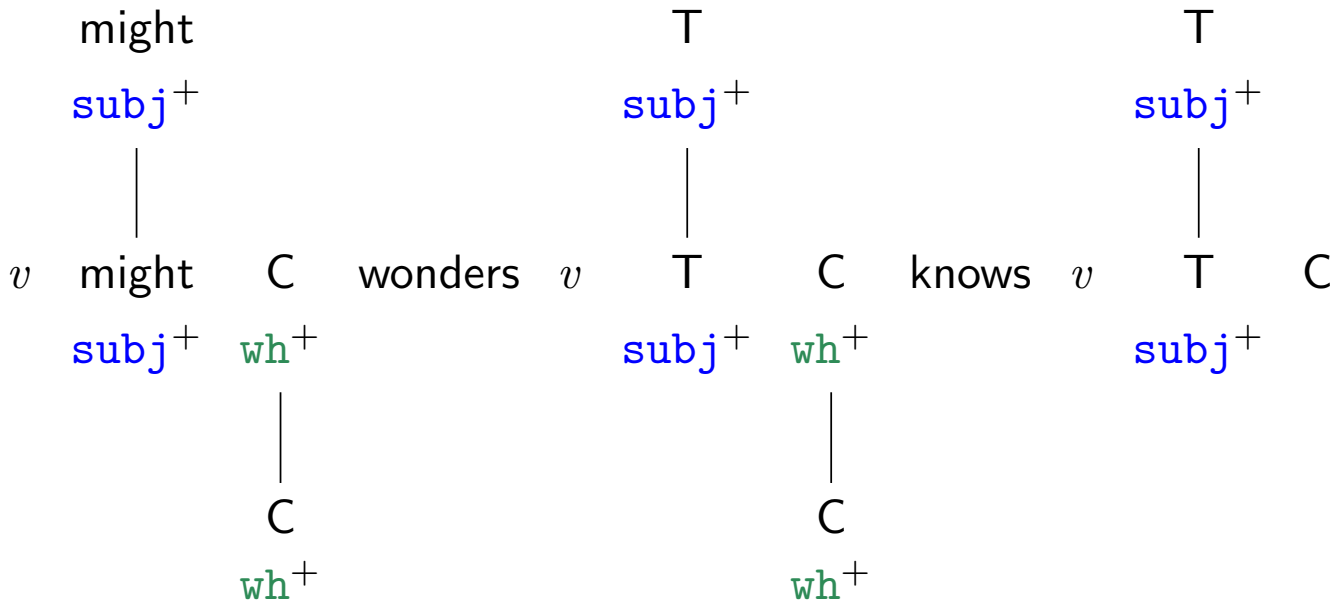
might
subj+
v might C wonders
subj+ wh+

T
subj+
v T C knows
subj+ wh+

T
subj+
v T C
subj+

A slightly larger example





What is so simple about tier-based reverse 1-definite?

String sets that are tier-based reverse 1-definite
require barely any cognitive abilities to verify well-formedness:

1. **Streaming**
   Read in symbols one after the other.
2. **Discrimination/Tier projection**
   Ignore symbols you don't care about.
3. **No memory**
   You don't need to represent the full tier.
   You don't need to remember anything.
   Look at each symbol one after the other,
   until you see the first symbol you care about.
   You are done.

Surprisingly simple, **surprisingly useful**

What is so simple about tier-based reverse 1-definite?

String sets that are tier-based reverse 1-definite
require barely any cognitive abilities to verify well-formedness:

1. **Streaming**
   Read in symbols one after the other.
2. **Discrimination/Tier projection**
   Ignore symbols you don't care about.
3. **No memory**
   You don't need to represent the full tier.
   You don't need to remember anything.
   Look at each symbol one after the other,
   until you see the first symbol you care about.
   You are done.

Surprisingly simple, **surprisingly useful**

Island constraints

- A cognitive/computational system that can handle the Ceiling Constraint, can also produce **island effects**!
- How? Just **project crud**...

Example: projecting *because* creates an adjunct island effect

- **Who did you complain [because Alex might kiss t?]*

kiss v might because complain v T did $\texttt{wh}^+$

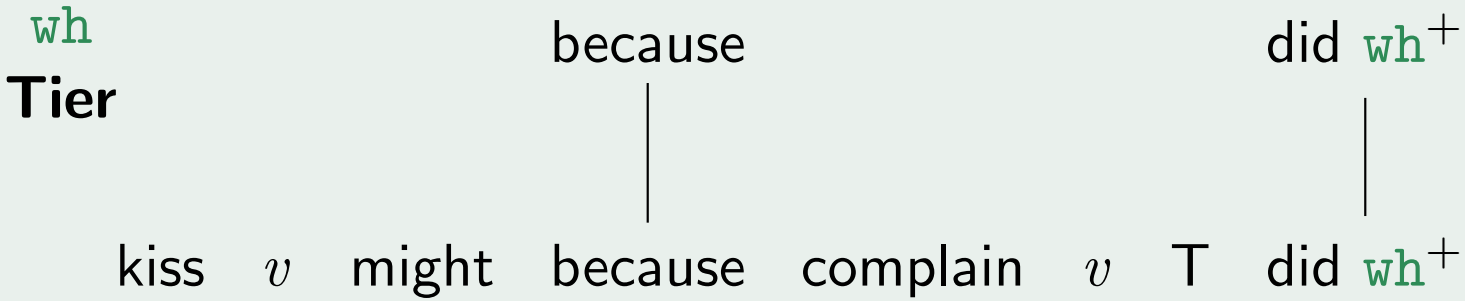
- The first symbol on the wh-tier doesn't have $\texttt{wh}^+$ $\Rightarrow$ the Ceiling Constraint is violated

Island constraints

- A cognitive/computational system that can handle the Ceiling Constraint, can also produce **island effects**!
- How? Just **project crud**...

Example: projecting *because* creates an adjunct island effect

- **Who did you complain [because Alex might kiss* t*?]*



- The first symbol on the wh-tier doesn't have $\texttt{wh}^+$ $\Rightarrow$ the Ceiling Constraint is violated

What just happened?

Ontological entailment from cognitive lower bounds

- We know that **X** exists and requires cognitive resource **R**.
- Cognitive resource **R** also allows computing **Y**.
- Hence we shouldn't be surprised to find **Y**.

(The argument is the exact opposite of the Ferris wheel fallacy.)

- The existence of (some) island constraints is unsurprising.
- A computational device that can handle movement also has the means to limit movement via islands.

Which island constraints work like this?

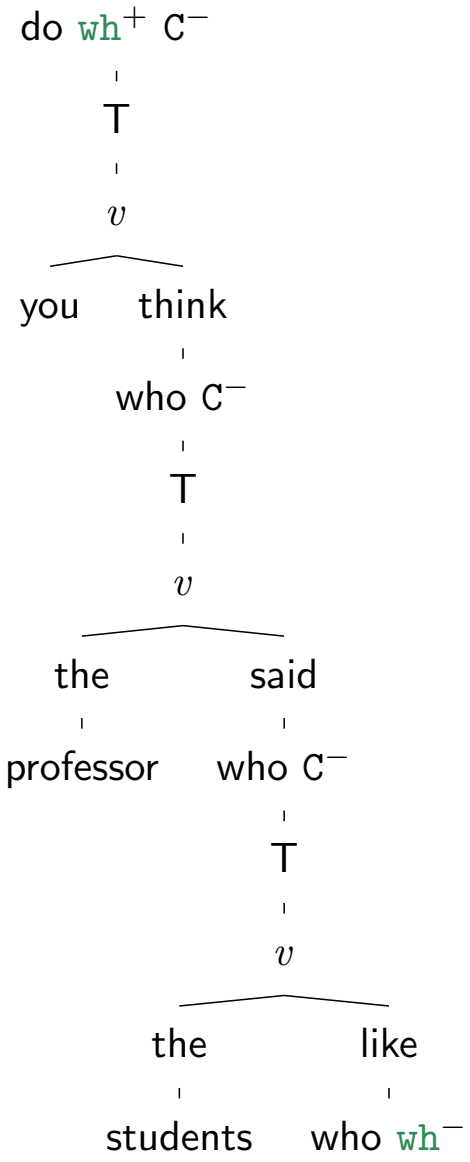
- **Adjunct island constraint**
  project a node if is the head of an adjunct
  presumably that is easy to determine
- **Complex NP constraint**
  project any N that selects C
  i.e. any node with both $\texttt{N}^{-}$ and $\texttt{C}^{+}$
- And many other phenomena actually are island-like.

German wh-copying

(1) Wen glaubst du [wen der Professor gesagt hat [wen
who think.2s you [who the professor said has [who
die Studenten mögen t]]?
the students like]]
'Who do you think that the professor said that the students like?'

- The copy movement analysis has tons of problems.
  illicit with full wh-phrases, copy conversion, interaction with *dass*, and much more; see Pankau (2013)
- Our computational approach allows for German wh-copying with no additional machinery.

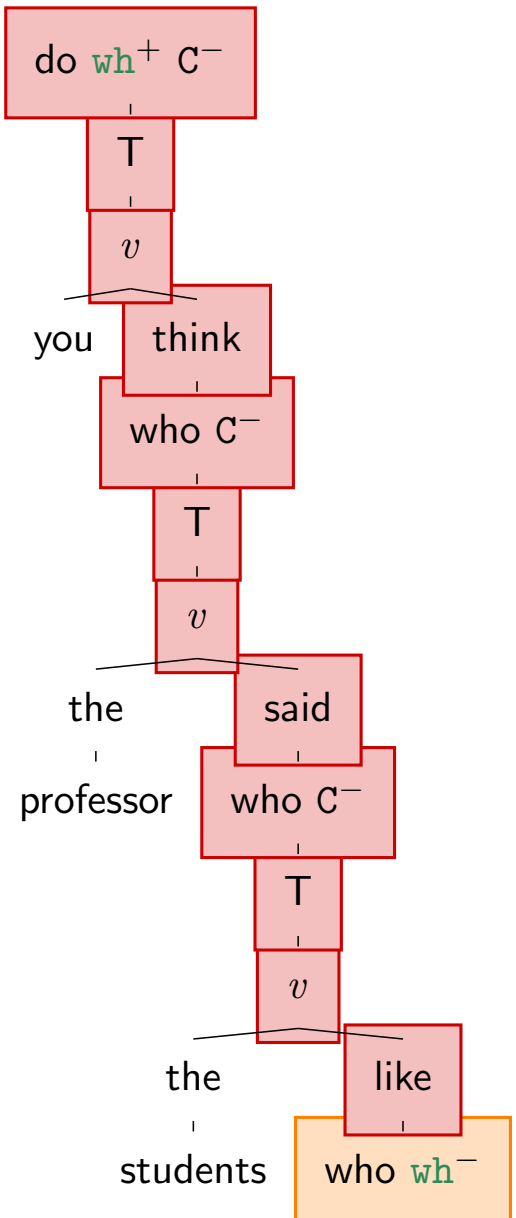
German wh-copying example



- **Ceiling Constraint**
  checked as usual
- **Assumption**
  wh-copies are some functional head;
  let's go with C for convenience
- **Tier-based reverse 1-definite constraints**

Node	Project...	Start with...
who wh−	all nodes with C−	wh+ or who C−
what wh−	all nodes with C−	wh+ or what C−
who C−	all nodes with C−	wh+ or who C−
what C−	all nodes with C−	wh+ or what C−

German wh-copying example

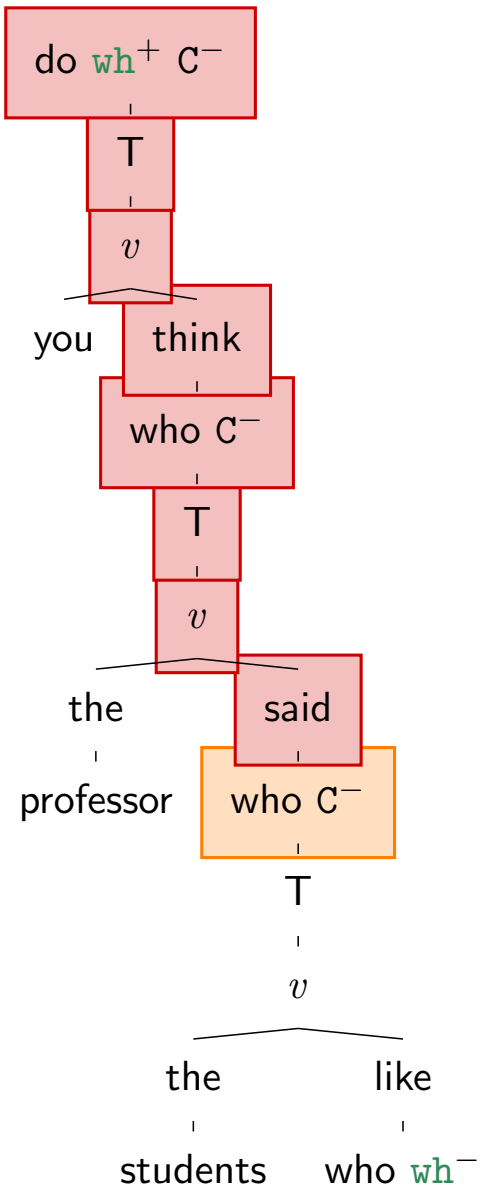


- **Ceiling Constraint**
  checked as usual
- **Assumption**
  wh-copies are some functional head;
  let's go with C for convenience
- **Tier-based reverse 1-definite constraints**

Node	Project...	Start with...
who wh−	all nodes with C−	wh+ or who C−
what wh−	all nodes with C−	wh+ or what C−
who C−	all nodes with C−	wh+ or who C−
what C−	all nodes with C−	wh+ or what C−

who C− who C− do wh+ C−

German wh-copying example

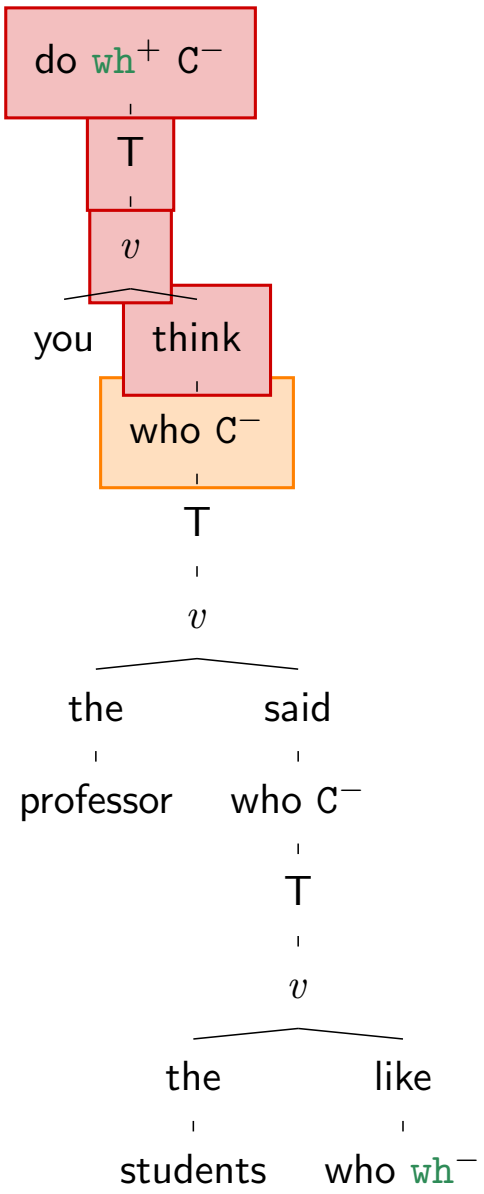


- **Ceiling Constraint**
  checked as usual
- **Assumption**
  wh-copies are some functional head;
  let's go with C for convenience
- **Tier-based reverse 1-definite constraints**

Node	Project...	Start with...
who wh−	all nodes with C−	wh+ or who C−
what wh−	all nodes with C−	wh+ or what C−
who C−	all nodes with C−	wh+ or who C−
what C−	all nodes with C−	wh+ or what C−

who C− do wh+ C−

German wh-copying example



- **Ceiling Constraint**
  checked as usual
- **Assumption**
  wh-copies are some functional head;
  let's go with C for convenience
- **Tier-based reverse 1-definite constraints**

Node	Project...	Start with...
who wh−	all nodes with C−	wh+ or who C−
what wh−	all nodes with C−	wh+ or what C−
who C−	all nodes with C−	wh+ or who C−
what C−	all nodes with C−	wh+ or what C−

do wh+ C−

Dealing with *dass* 'that'

- Wh-copying can skip some C-heads at first.

(2) a. **Who** do you think **who** the professor said **who** the students like t?
b. **Who** do you think **who** the professor said *that* the students like t?
c. **Who** do you think *that* the professor said *that* the students like t?

- But once you do wh-copying, you can't go back to *dass*.

(3) * **Who** do you think *that* the professor said **who** the students like t?

- Again unsurprising, requires **just a minor tweak**

Node	Project...	Start with...
who wh$^-$	all nodes with C$^-$	wh$^+$ or who C$^-$
what wh$^-$	all nodes with C$^-$	wh$^+$ or what C$^-$
who C$^-$	all nodes with C$^-$	wh$^+$ or who C$^-$
what C$^-$	all nodes with C$^-$	wh$^+$ or what C$^-$

(Exercise: in some dialects you can mix wh-copies and *dass*, and that is again unsurprising; what do we need to change above?)

Dealing with *dass* 'that'

- Wh-copying can skip some C-heads at first.

(2) a. **Who** do you think **who** the professor said **who** the students like t?
b. **Who** do you think **who** the professor said *that* the students like t?
c. **Who** do you think *that* the professor said *that* the students like t?

- But once you do wh-copying, you can't go back to *dass*.

(3) * **Who** do you think *that* the professor said **who** the students like t?

- Again unsurprising, requires **just a minor tweak**

Node	Project...	Start with...
who wh$^-$	all nodes with C$^-$ except *that*	wh$^+$ or who C$^-$
what wh$^-$	all nodes with C$^-$ except *that*	wh$^+$ or what C$^-$
who C$^-$	all nodes with C$^-$	wh$^+$ or who C$^-$
what C$^-$	all nodes with C$^-$	wh$^+$ or what C$^-$

(Exercise: in some dialects you can mix wh-copies and *dass*, and that is again unsurprising; what do we need to change above?)

Comparison to wh copy movement analysis

Our view of German wh-copying. . .

- . . . likens it to island effects
  mis-agreeing C-heads are islands
- . . . explains why full wh-phrases can't be copied
  the copies are functional heads, not phrases
- . . . allows for copy conversion, e.g. *wo* 'where' as the wh-copy of *in welchem Haus* 'in which house'
  we can allow *wo* as the first symbol for *in wh*
- . . . captures the dialectal variation with respect to *dass* 'that'
  minor differences in what projects for what types of nodes
- . . . requires **no extra machinery whatsoever**

German wh-copying ≈ extraction morphology

- **Extraction morphology** refers to cases where a node's morphological realization is affected by other material moving across it
  - aL/go alternation in Irish
  - Wolof u-chains
  - Chamorro wh-agreement
  - Duala *no*-marking
  - Ewe subject pronoun choice
  - ⋮
- **Spoilers**: these are **variants of German wh-copying**
  or rather, German wh-copying is extraction morphology

Georgi's typology of extraction morphology

- Georgi (2017) identifies a Minimalist 2-by-2 typology depending on whether extraction morphology appears on
  - the final landing site: Yes/No
  - the intermediate landing sites: Yes/No
- We correctly predict these to be independent parameters.
  - agreement on final landing site:
    can be done as part of the Ceiling Constraint
  - agreement on intermediate landing sites:
    separate constraints like German wh-copying

An interesting quirk: Duala no-marking

- In Duala (Niger-Congo), the extraction morphology appears on a head that the mover does not move through.
- *no* appears on T of the clause with the final landing site

(4) Kalati$_i$ nde Kuo a bodi *(**no**) nu moto t_i kiele.
book Foc Kuo 3sg give no that man yesterday
'It's a book Kuo gave to that man yesterday.' (Epée 1976)

- Problem for movement-based accounts, but not for us.

Node	**Project...**	**Start with...**
wh$^-$	all T-*no* and wh$^+$	T-*no*
T-*no*	all T, T-*no*, wh$^-$, and wh$^+$	wh$^+$

Ewe subject alternation

- In Ewe (Niger Congo),
  the 3rd person subject *é* becomes *wò*
  if a wh-phrase moves across.

(5) Kofi biɛ [$_{\text{CP}}$ be lamata$_i$ **wò**/*é fo Kɔsi t_i].
Kofi asked C why he hit Kɔsi
'Kofi asked why he hit Kɔsi.'

- We're stuck. Do you know why?

Ewe subject alternation

- In Ewe (Niger Congo),
  the 3rd person subject *é* becomes *wò*
  if a wh-phrase moves across.

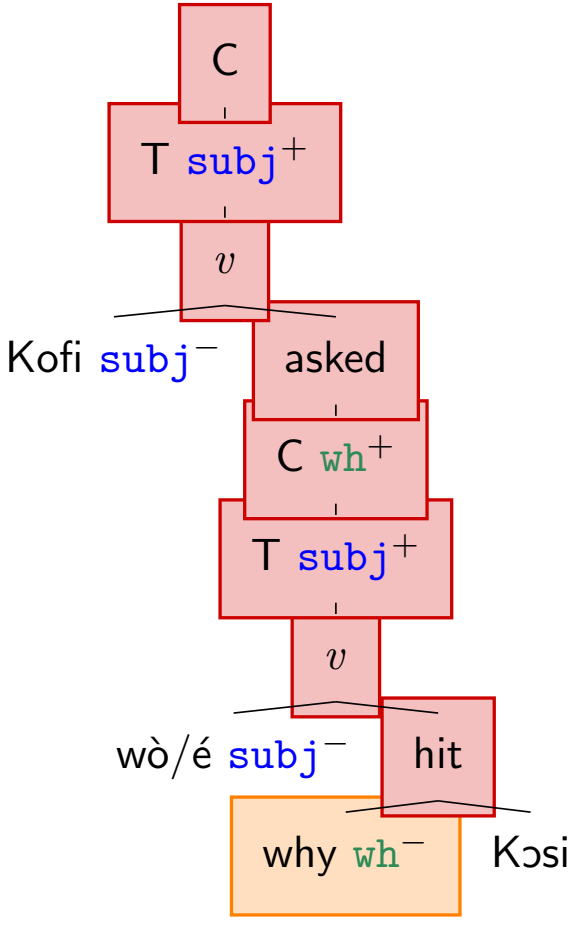
(5)

Kofi	biɛ	[$_{\text{CP}}$ be	lamata$_i$	**wò**/*é	fo
Kofi	asked	C	why	he	hit
Kɔsi	t_i].				
Kɔsi					

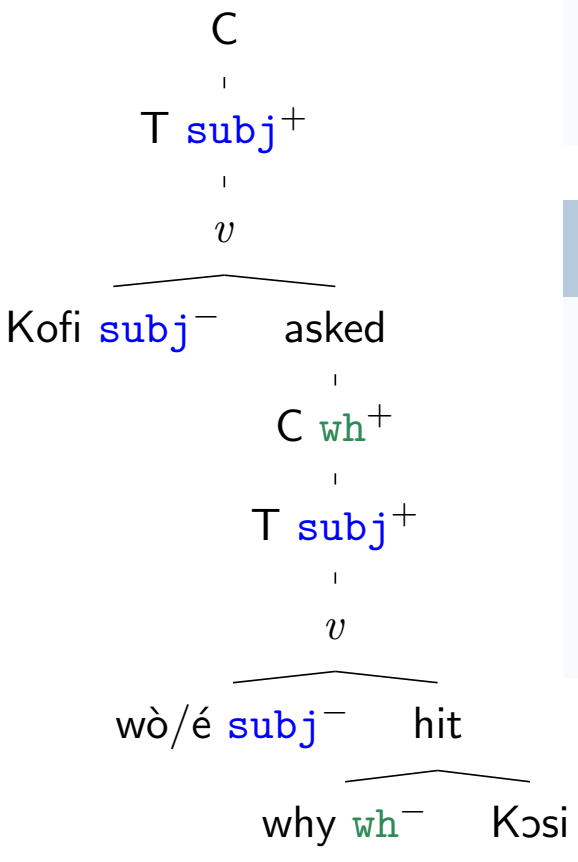
'Kofi asked why he hit Kɔsi.'

- We're stuck. Do you know why?

You can't constrain what you can't see!



A new string type: c[ommand]-strings



Definition (a[ncestor]-string; simplified)

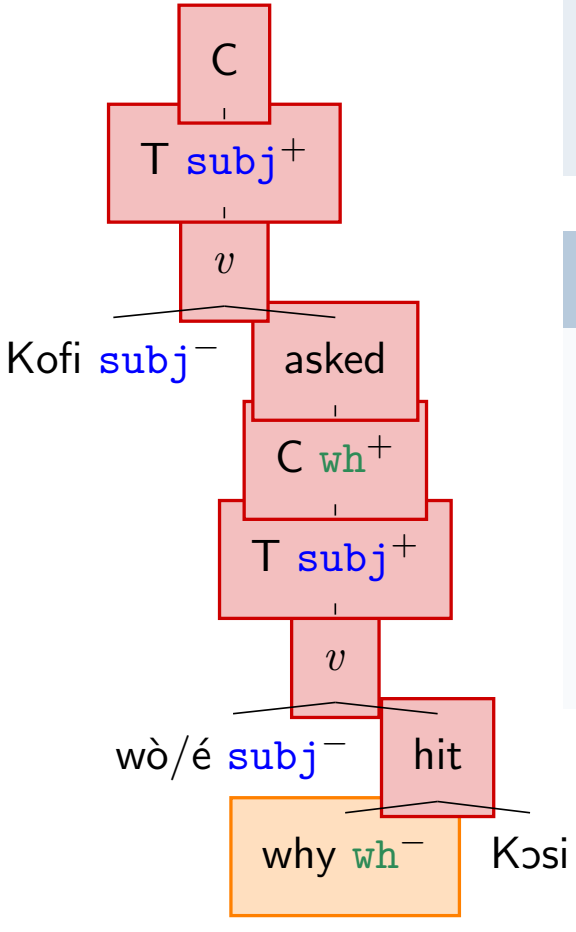
The **a-string** of **n** consists of all the nodes that are derivationally ordered after **n**.

Definition (c[ommand]-string; simplified)

The **c-string** of **n** consists of all the nodes that are either

- in the a-string of **n**, or
- left siblings of a node in the a-string of **n**.

A new string type: c[ommand]-strings



Definition (a[ncestor]-string; simplified)

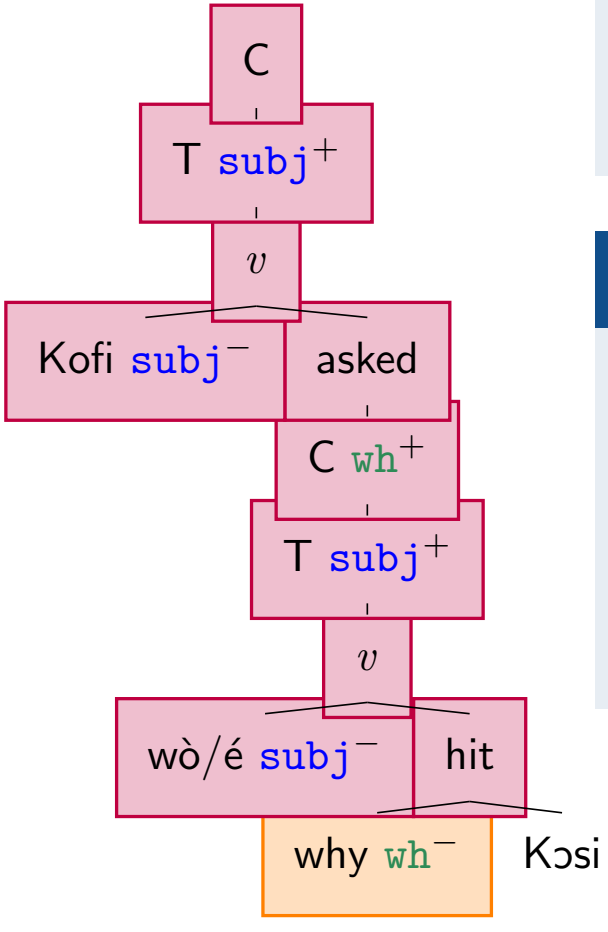
The **a-string** of **n** consists of all the nodes that are derivationally ordered after **n**.

Definition (c[ommand]-string; simplified)

The **c-string** of **n** consists of all the nodes that are either

- in the a-string of **n**, or
- left siblings of a node in the a-string of **n**.

A new string type: c[ommand]-strings



Definition (a[ncestor]-string; simplified)

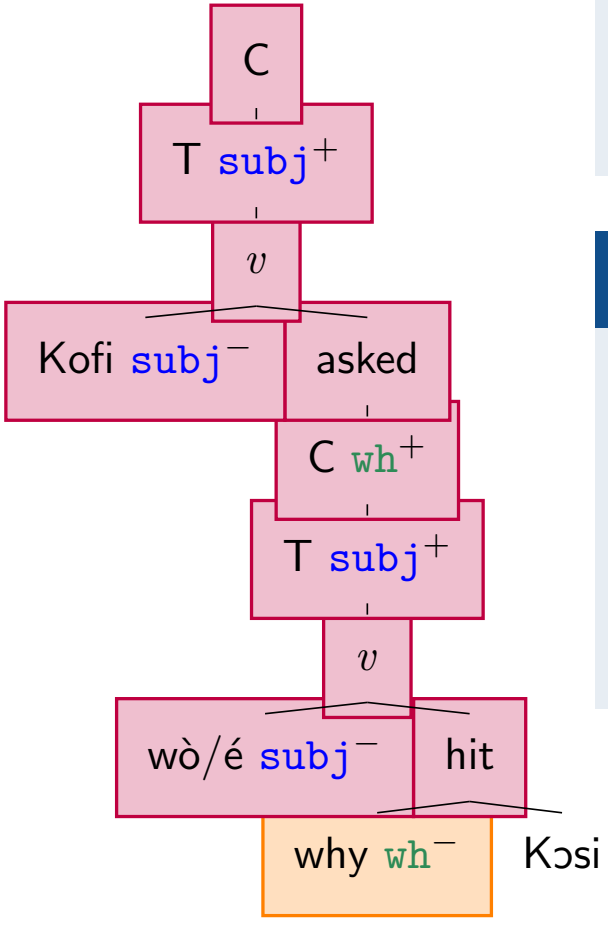
The **a-string** of **n** consists of all the nodes that are derivationally ordered after **n**.

Definition (c[ommand]-string; simplified)

The **c-string** of **n** consists of all the nodes that are either

- in the a-string of **n**, or
- left siblings of a node in the a-string of **n**.

A new string type: c[ommand]-strings



Definition (a[ncestor]-string; simplified)

The **a-string** of **n** consists of all the nodes that are derivationally ordered after **n**.

Definition (c[ommand]-string; simplified)

The **c-string** of **n** consists of all the nodes that are either

- in the a-string of **n**, or
- left siblings of a node in the a-string of **n**.

Node	Project...	Start with...
wh−	wh+ and *é*	wh+

é acts like an island for wh-movers

Relativized minimality

- A new string type just for Ewe? No, for a lot of stuff!
  Binding! Agree! But we don't have time...

(6) a. Who bought what?
b. * What did who buy?

- Relativized minimality = **island effect over c-strings**

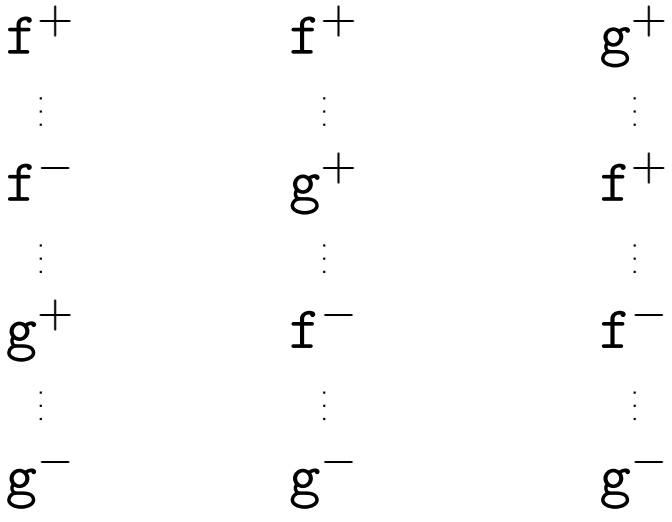
Node	**Project...**	**Start with...**
$\texttt{wh}^-$	$\texttt{wh}^+$ and all potential carriers of $\texttt{wh}^-$	$\texttt{wh}^+$

c-string of *who* in (6a): v T C$\texttt{wh}^+$ Tier: C$\texttt{wh}^+$
c-string of *what* in (6b): buy who v T C$\texttt{wh}^+$ Tier: who C$\texttt{wh}^+$

A minor extension for freezing effects

- **Freezing**: Once an XP has started moving,
  no YP contained in XP may move out of XP.
  a relativized version like UCOOL is empirically more adequate,
  but works the same for our purposes
- Freezing can be done over a-strings,
  but it requires a larger window: **tier-based reverse 2-definite**

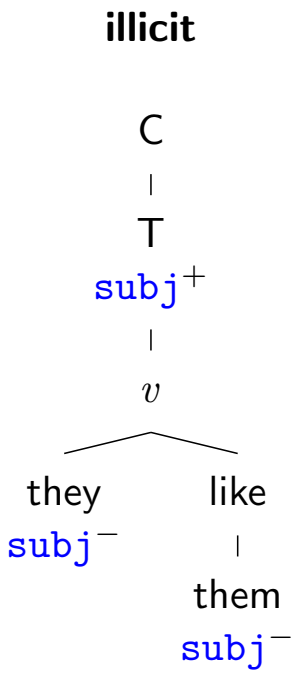
allowed **allowed** **forbidden**



Node	Project...	Start with...
g^-	f^-, f^+, g^+	not f^-f^+

Size-2 windows for movement

- Freezing's larger window isn't that special, we need it for movement, too.
- **The Ceiling Constraint**
  Every $\texttt{f}^-$ needs an $\texttt{f}^+$
- **The Foundation Constraint**
  Every $\texttt{f}^+$ is matched with exactly one $\texttt{f}^-$



A brief sketch for T$\texttt{subj}^+$

- Do a recursive descent traversal from T$\texttt{subj}^+$, but don't go below any $\texttt{subj}^-$.
- Resulting string: v they$\texttt{subj}^-$ like them$\texttt{subj}^-$
- The $\texttt{subj}^-$ tier must have length 1.
  Checking this requires a window of size 2.

Overview of string-types and window size

Constraint	**string**	**size**
Ceiling Constraint	a	1
Foundation Constraint	rec.desc.	2
Strong islands	a	1
German wh-copying	a	1
	rec.desc.	1
Extraction morphology	a	1
	rec.desc.	1
Ewe subject alternation	c	1
Principle A	$\sim$c	1
Long-distance reflexives	$\sim$c	1
Freezing	a	2
Ban on Improper movement	a	1
Omnivorous agreement	rec.desc.	2

A mathematical theorem: from local to global

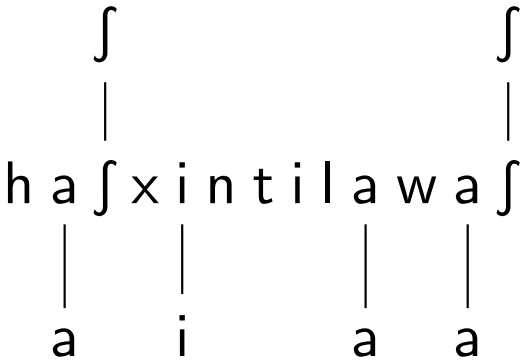
- Isn't it taxing (and redundant) for syntax to compute these strings for each node?
- No! For each string type τ we can convert this to a **single constraint** that directly defines the set of all well-formed τ-strings.

Toy example without tiers

- Suppose our nodes can only be x or y.
- The a-string of x must start with y.
- The a-string of y must start with x or contain nothing at all.
- Then the set of well-formed a-strings is exactly the set of strings that may only contain the following bigrams:
  - xy (x followed by y)
  - yx (y followed by x)
  - $y\rtimes$ (y followed by the right string edge)

Syntax as multi-phonology

- For each string type τ, the converted constraint is **multi tier-based strictly local (MTSL)**.
- This type of constraint is also **very common in phonology**.



- **Syntax as multi-phonology**
  - Phonology: building a single MTSL string
  - Syntax: building many intertwined MTSL strings

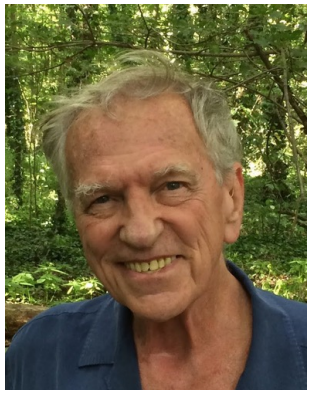
Open ends

- Connections to other work?
  Kayne's unambiguous paths, path equations in LFG, trees as path languages
- Some phenomena don't fit this view.
  parasitic gaps, coordinate structure constraint
- Many phenomena I haven't had time to look at yet.
  Anybody looking for a research project?

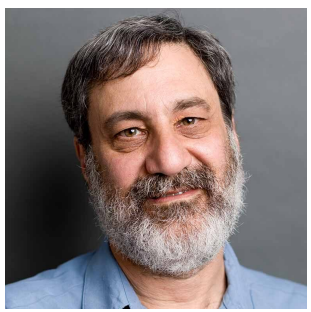
Conclusion: don't lock yourself into a single viewpoint

- Don't just think in terms of trees, strings are very insightful!
- A variety of syntactic phenomena belong to a very simple class of constraints over particular types of strings.
  a-strings, c-strings, rec.desc.-strings
- Computation provides an explanation for why these phenomena can occur in the first place.
- The view is much more abstract than regular syntax, making it easier to hook into cognitive science, NLP, ...
- When go from the node-based perspective back to a more global one, we get **syntax as multi-phonology**.

If you can't say something two ways,
you can't say it at all.
(Keenan and Moss 2017)



Edward Keenan



Larry Moss

References

Georgi, Doreen. 2017. Patterns of movement reflexes as the result of the order of merge and agree. *Linguistic Inquiry* 48:585–626.

Keenan, Edward, and Larry Moss. 2017. *Mathematical structures in language*. Stanford: CSLI.

Lambert, Dakotah. 2022. *Unifying classification schemes for languages and processes with attention to locality and relativizations thereof* . Doctoral Dissertation, Stony Brook University.

Pankau, Andreas. 2013. *Replacing copies: The syntax of wh-copying in German*. Doctoral Dissertation, University of Utrecht, Utrecht, Netherlands.

Poeppel, David, and David Embick. 2005. Defining the relation between linguistics and neuroscience. In *Twenty-first century psycholinguistics: Four cornerstones*, ed. Anne Cutler, 103–118. New York, NY: Routledge.