

November 12ish, 2019 — v1.1

Note: Revision 1.1 adds a bunch of X' nodes in the trees that had been left out.

1 X-bar theory

- (1) a. $XP \rightarrow WP XP$
- b. $XP \rightarrow XP WP$
- c. $XP \rightarrow (YP) X'$
- d. $X' \rightarrow X' WP$
- e. $X' \rightarrow WP X'$
- f. $X' \rightarrow X (ZP)$

- Phrases (XP) have a unique head (X). Features from X project to XP.
- Complement (YP) is sister to the head (X).
- Specifier (ZP) is sister to X' and daughter of XP.
- Phrases (WP) can be adjoined to phrases (XP). A hypothesis/policy decision
- Phrases (WP) can be adjoined to X' . A hypothesis/policy decision
- There is always an X' . A hypothesis/policy decision
- Branching need not be binary. A hypothesis/policy decision

2 Auxiliaries

- (2) I should eat.
- (3) I should have eaten.
- (4) I should be eating.
- (5) I should have been eating.

How do we do this? This is actually a kind of hard problem.

- (6) I should eat.
- (7) * I have should eaten.
- (8) * I have shoulden eat.
- (9) * I be should eating.
- (10) * I be shoulding eat.

They have to be in a specific order, but they don't always have to be there.

C-selection? Clumsy, baroque, seems wrong. How do we do it?

There seems to be an ordering that we can't capture with existing technology.

Wait, we had another one too.

(11) The big red hat.

(12) * The red big hat.

We don't know why those have an order either, it just seems to be something like:

(13) value > Size > Temperature > Age > Shape > Color > origin > Material

(14) This is made from beautiful small cold 3-year-old square blue Italian silver coins.

Something like.

So, let's say there's a similar kind of externally imposed ordering here;

(15) modal/to > have+en > be+ing > be+en > verb

(16) modal/to > perfective > progressive > passive > verb

(17) Sprouts should have been being eaten.

Now that we have established the order itself, how to draw this in a tree?

We said that modals are T. That worked ok. But what is *have*? Or *be*?

(18) I have a pencil.

(19) I have bought a pencil.

(20) I am tall.

(21) I am buying a pencil.

Arguably two kinds of *have* and *be*, one a real auxiliary, one a real verb.

These auxiliaries *have*, *be*, and in fact *do*, all are kind of intuitively "verby." (Why?)

What's the simplest way we could suppose something like *I have eaten lunch* could be drawn? (At first, ignoring the fact that the verb comes out in the form *eaten* instead of *eat* or *ate*).

There are weather verbs, intransitive verbs, transitive verbs, ditransitive verbs. Maybe this is a special kind of verb that specifically takes another VP as its complement. Those might be what we'd define to be the auxiliaries in fact, perhaps.

As for how to get *be eating* and *have eaten*, let's suppose this works kind of like how *ate* works. PAST *eat* = *ate*.

What is *be* in *be eating*? It's really a progressive marker.

Let's call it *be+ING*.

And *have*? *Have+EN*.

(22) a. I PAST eat lunch.

b. I ate lunch.

c. He PRES eat lunch.

d. He eats lunch.

(23) a. I PAST have+EN be+ING eat lunch.

b. I had been eating lunch.

c. He PRES have+EN eat lunch.

d. He has eaten lunch.

"Affix hopping"

(24) be+ING, V, [AUX, PROG], [+ _ VP]

(25) have+EN, V, [AUX, PERF], [+ _ VP]

External ordering:

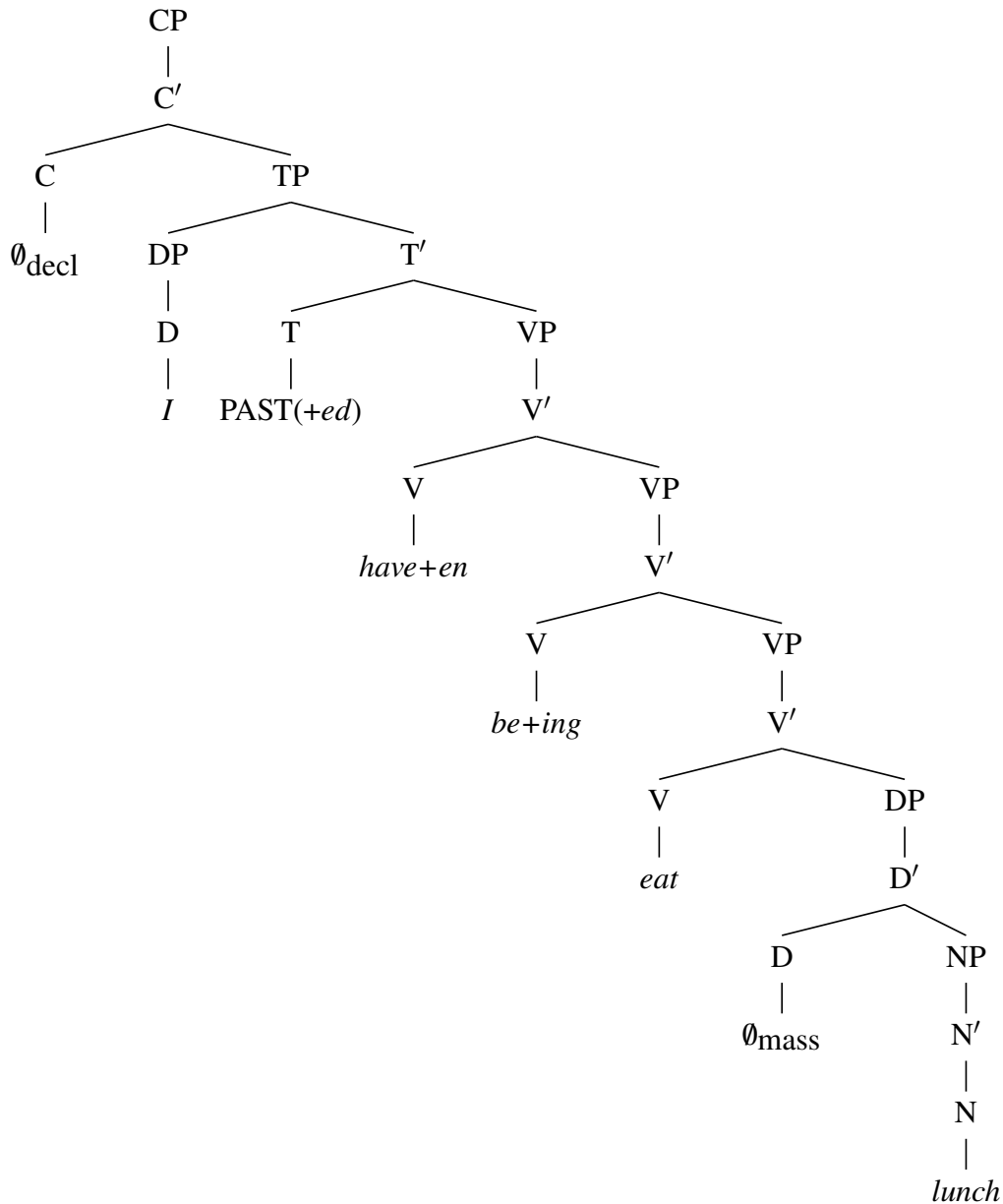
(26) Adjectives: value > Size > Temperature > Age > Shape > Color > origin > Material

(27) CP > TP > negation > modal/to > perfective > progressive > passive > verb

(28) DP > NP

(29) I PAST have-en be-ing eat lunch

(30) I had been eating lunch



3 Negation

We haven't really yet dealt with negation, but let's start. It's going to make things interesting.

- (31) I could eat lunch.
- (32) I could not eat lunch.
- (33) I can eat lunch.
- (34) I can not eat lunch.
- (35) I would eat lunch.
- (36) I would not eat lunch.
- (37) I will eat lunch.
- (38) I will not eat lunch.

The *not* comes between the modal and the verb.

We said the modal was in T. So, *not* must be between TP and VP.

One possibility is that it is just an adverb, attaching like *never*.

- (39) I could never eat lunch.
- (40) I could always eat lunch.
- (41) I could happily eat lunch.
- (42) I can never eat lunch.
- (43) I would never eat lunch.
- (44) I will never eat lunch.

But there's a funny difference between *not* and adverbs like *never*—if we just have a verb and no auxiliaries, no modals, we get *do*-support with *not* but not with *never*.

- (45) I never eat lunch.
- (46) I do not eat lunch.

They behave differently. It still seems like *not* is between TP and VP, but if there is not something else in T, we stick *do* in there. A meaningless verb. But there seemingly to carry tense and agreement.

Since we've already supposed that the auxiliaries are kind of stacked VPs, maybe we can just treat negation this way as well. It's not really quite as "verby" as auxiliaries, so we'll make it its own category.

- (47) *not*, Neg, [+ _ VP]

And we also have to suppose that T has the possibility of taking NegP as its complement.

- (48) PAST, T, [+ _ VP], [+ _ NegP]

Cool.

But, wait, consider:

- (49) I have not been eating lunch.
- (50) I was not eating lunch.
- (51) I should not have been eating lunch.

If *should* is in T, where are *have* and *was* in the earlier examples? Do we want to say that Neg can be between any two auxiliaries maybe?

- (52) She should not have been eating lunch, should she?
 (53) * She should have not been eating lunch, should she?
 (54) * She should have been not eating lunch, should she?
 (55) She should not have been not eating lunch, should she?

On balance, probably not. Also, look again at *can/could*, *may/might*, *shall/should*. Seems a bit like a present/past distinction, like in *eat/ate*.

How can we do this? We have an order; any auxiliaries present must be in this order:

- (56) Modal have-perf be-prog be-passive verb

But when there is *not*, it seems that the first one we have appears to the left of *not*. And if we don't have any auxiliaries/modals, then we put *do* to the left of *not*.

Suppose that we have an ordering like CP, TP, NegP, VP. Meaning that if there is a NegP, it is higher in the clause than any VPs.

The generalization seems like this: The highest auxiliary moves over Neg to T. If there is no auxiliary, then *do* is inserted. Main verbs to do not move up to T.

The PAST/PRES in T will determine the ending of the thing that moves up to T.

And so we have *movement*. Movement of the auxiliary to T. This makes sense of the pattern.

We need a way to formalize this at this point, maybe something like:

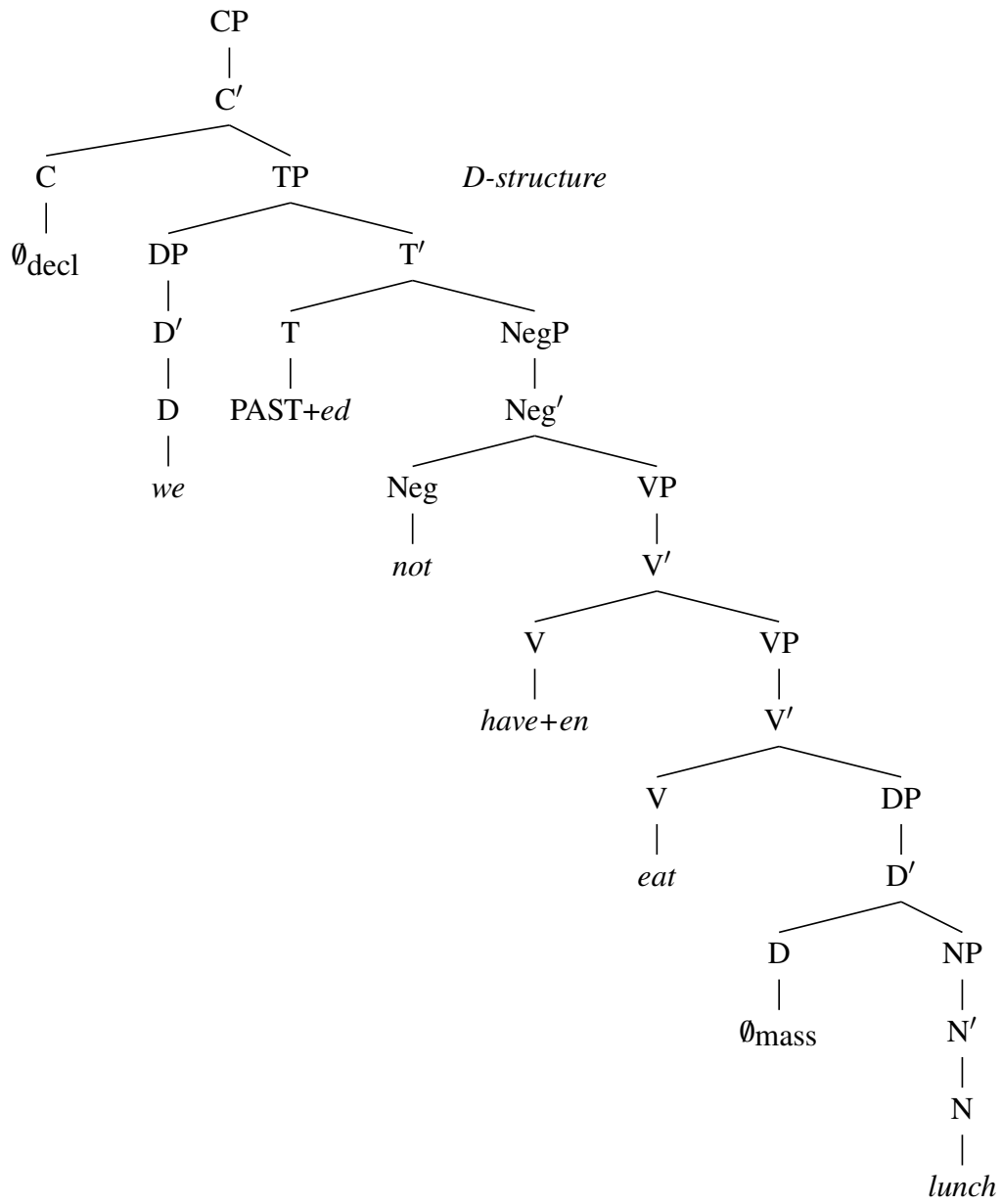
- (57) [... T Y [VP ... V ...]] → [... T+V Y [VP ... _ ...]] if V is [+AUX]

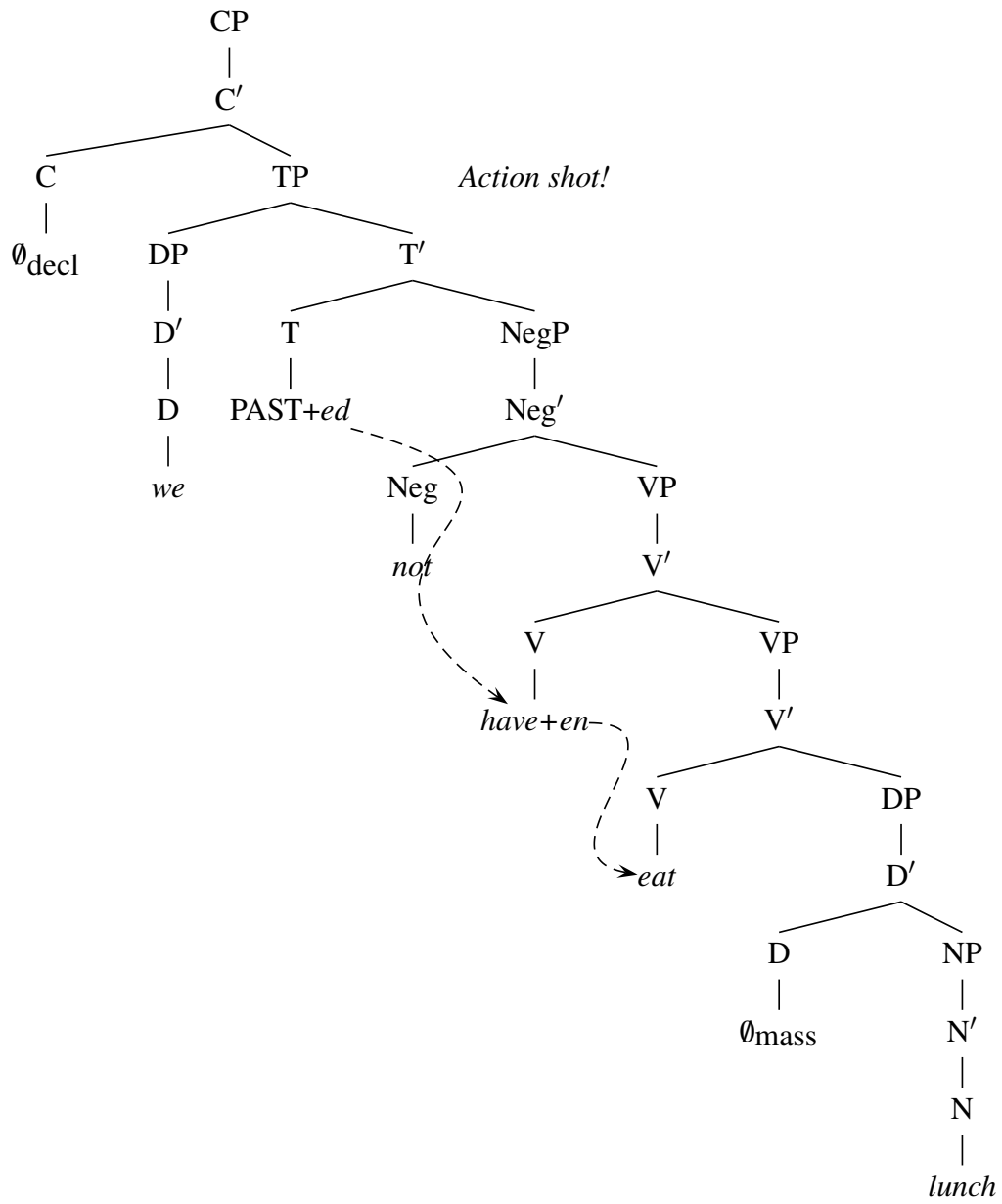
That will do for now.

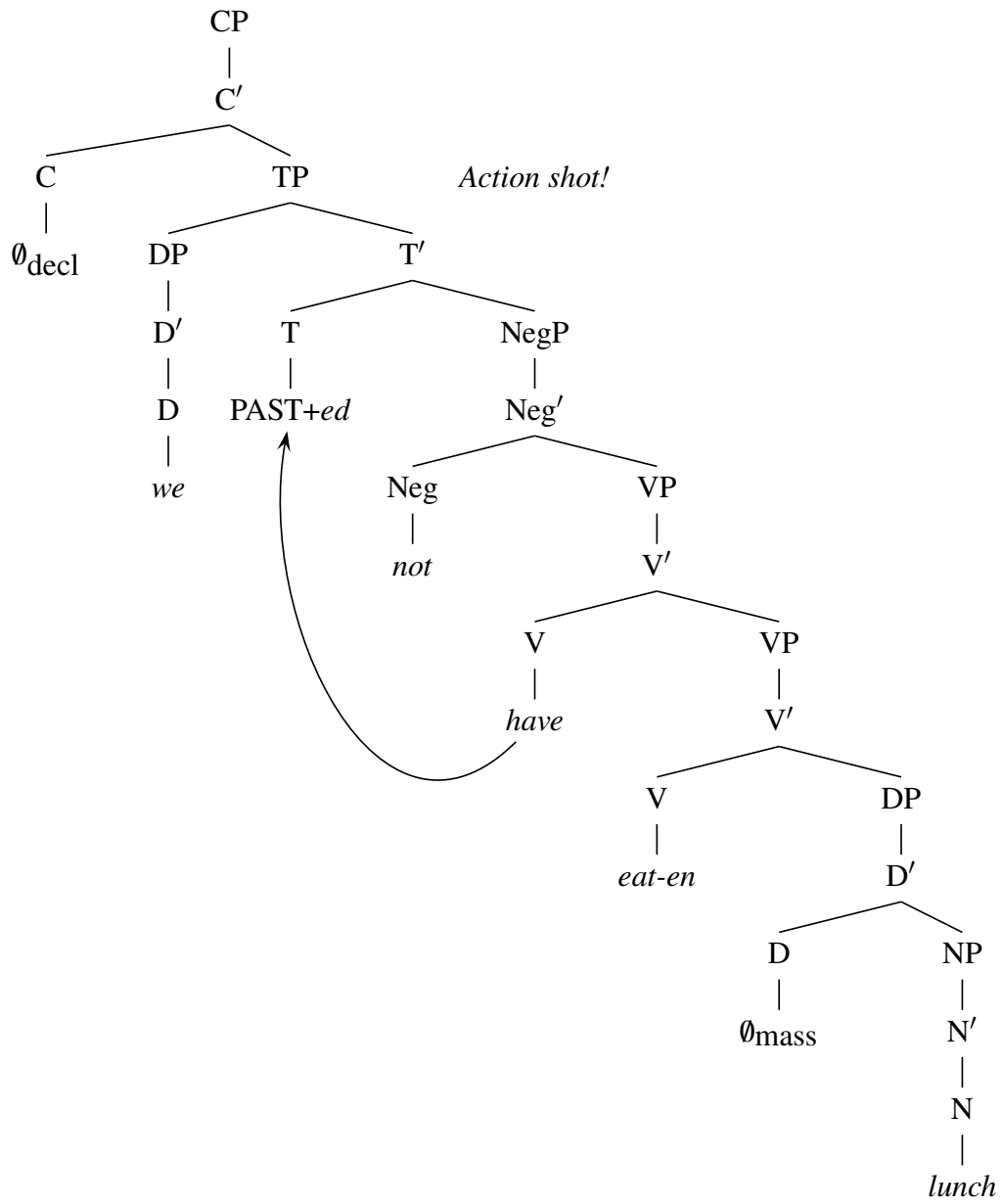
When do you get *do*?

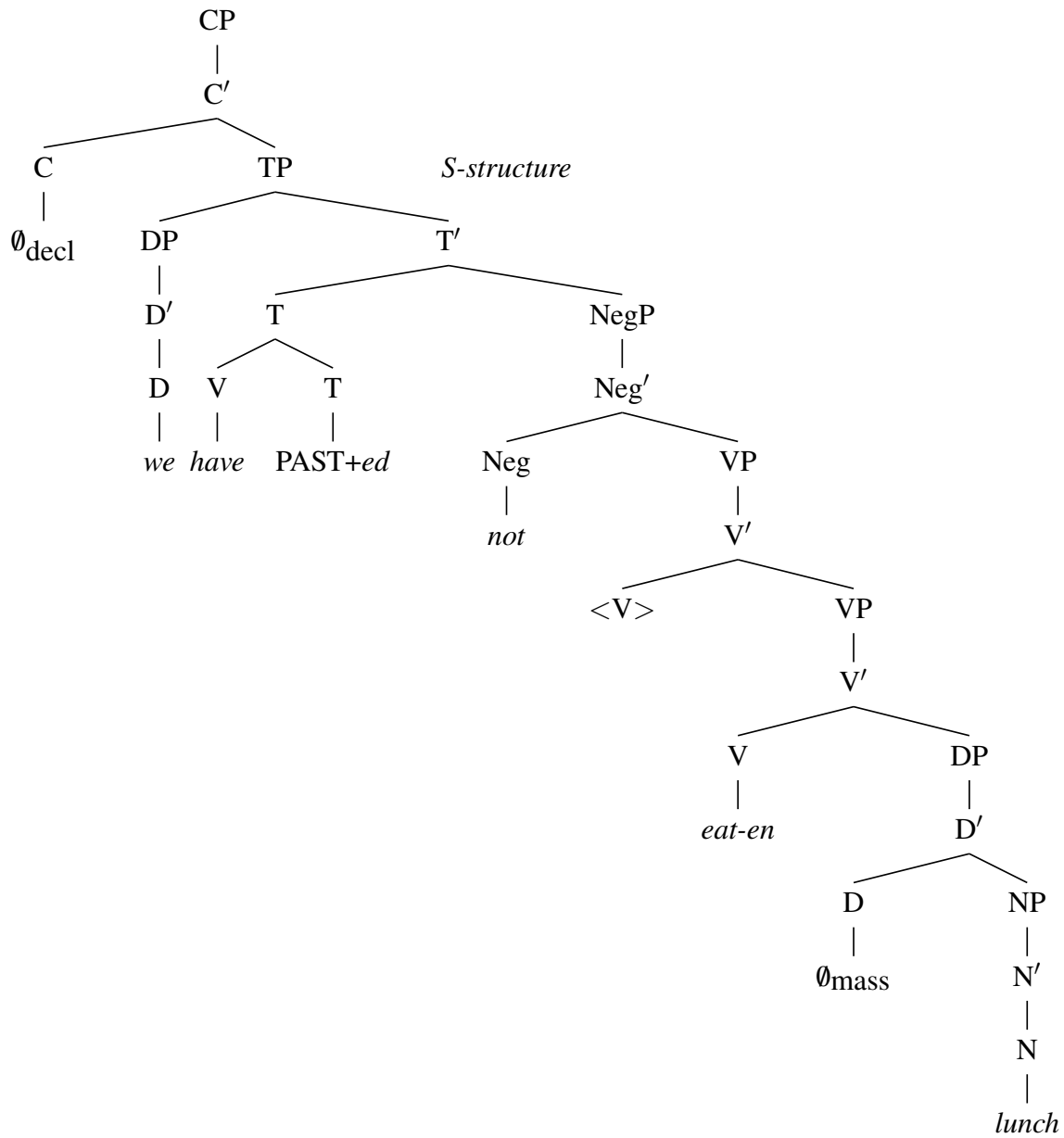
Basically if Y is there (and it can really only be Neg). For an adverb, Y is not there.

Cross-linguistic note: French can be said to move all V (not just [+AUX] V) to T, this is a parametric difference between the languages.



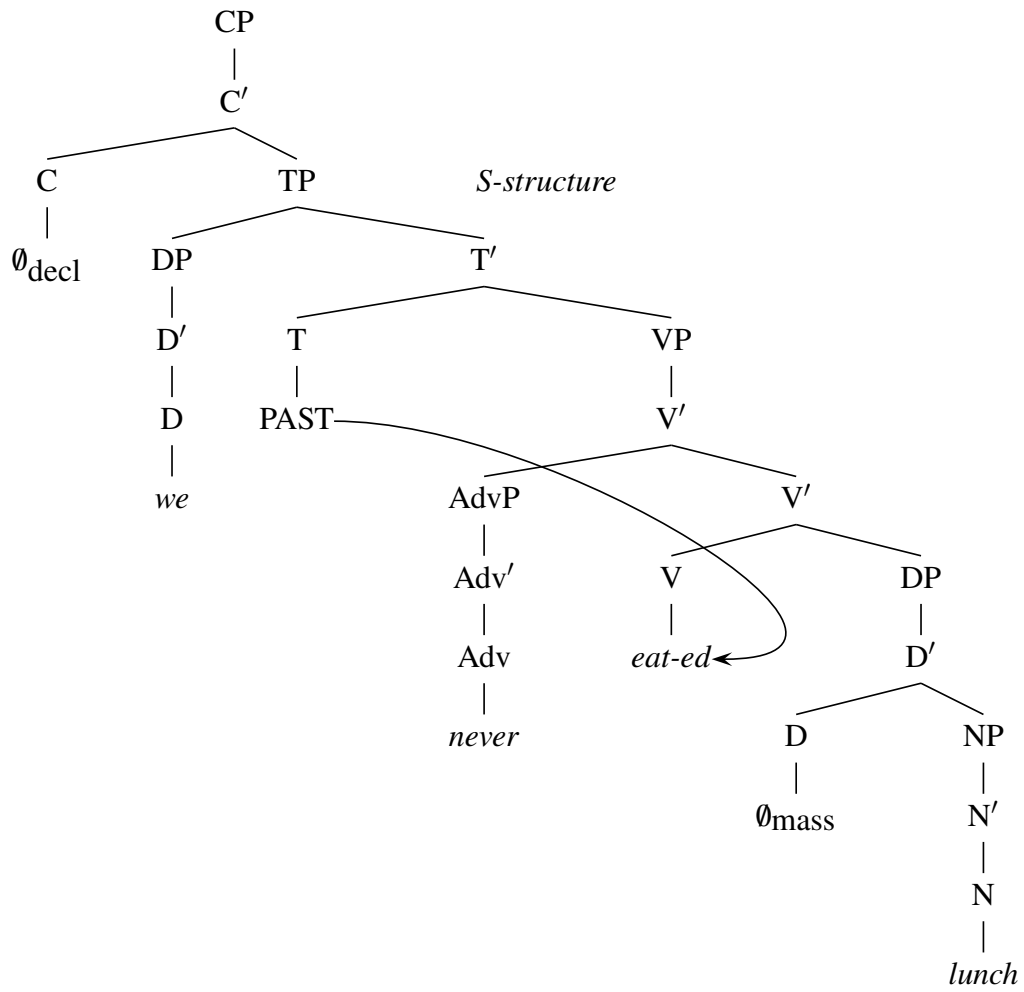


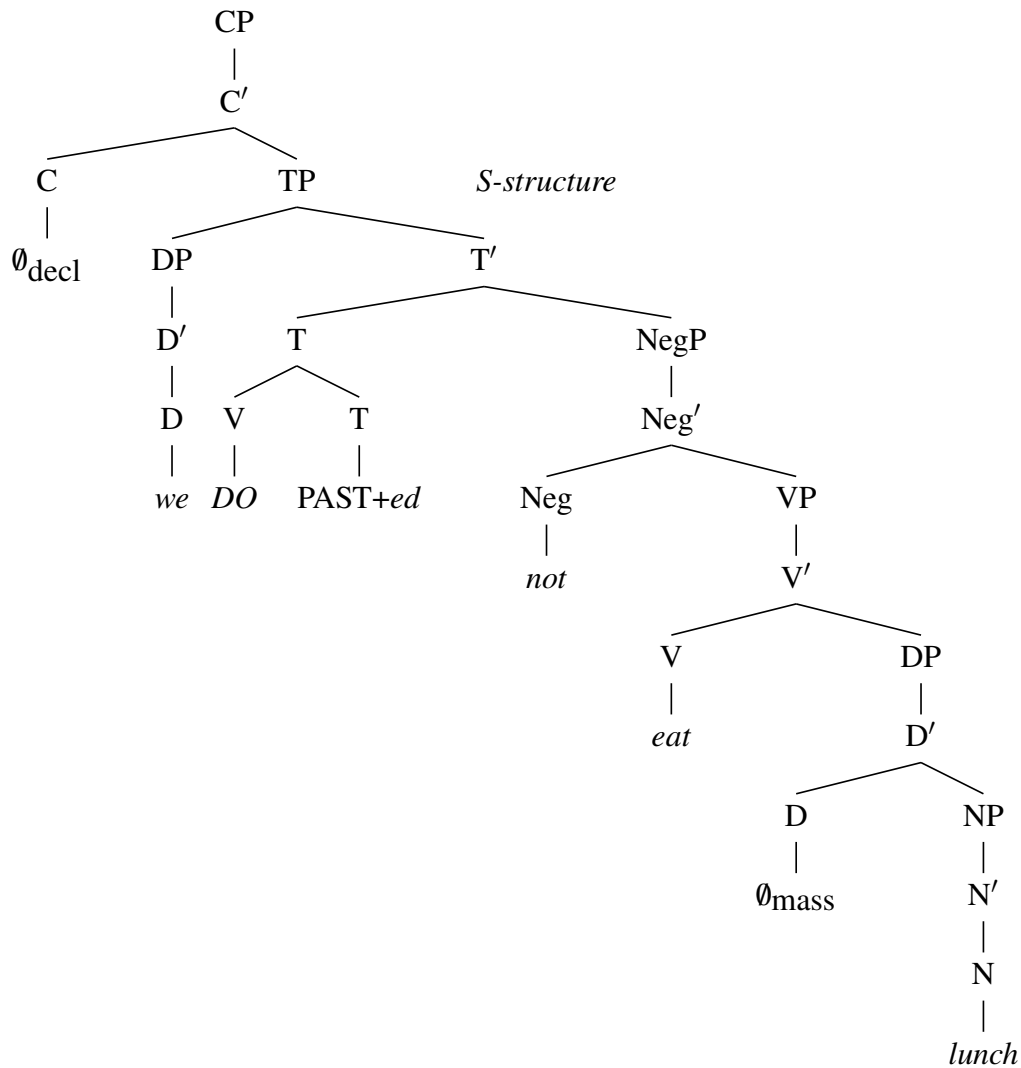




(58) I never eat lunch.

(59) I do not eat lunch.





Generalization? T is “stranded” if its sister is not headed by V. (That is, not a VP.)

NegP is not a VP. Adverbs attach to VPs and leave them VPs.

Auxiliaries (modals, *have*, *be*) move to T (even over Neg) so T is not stranded in that case.

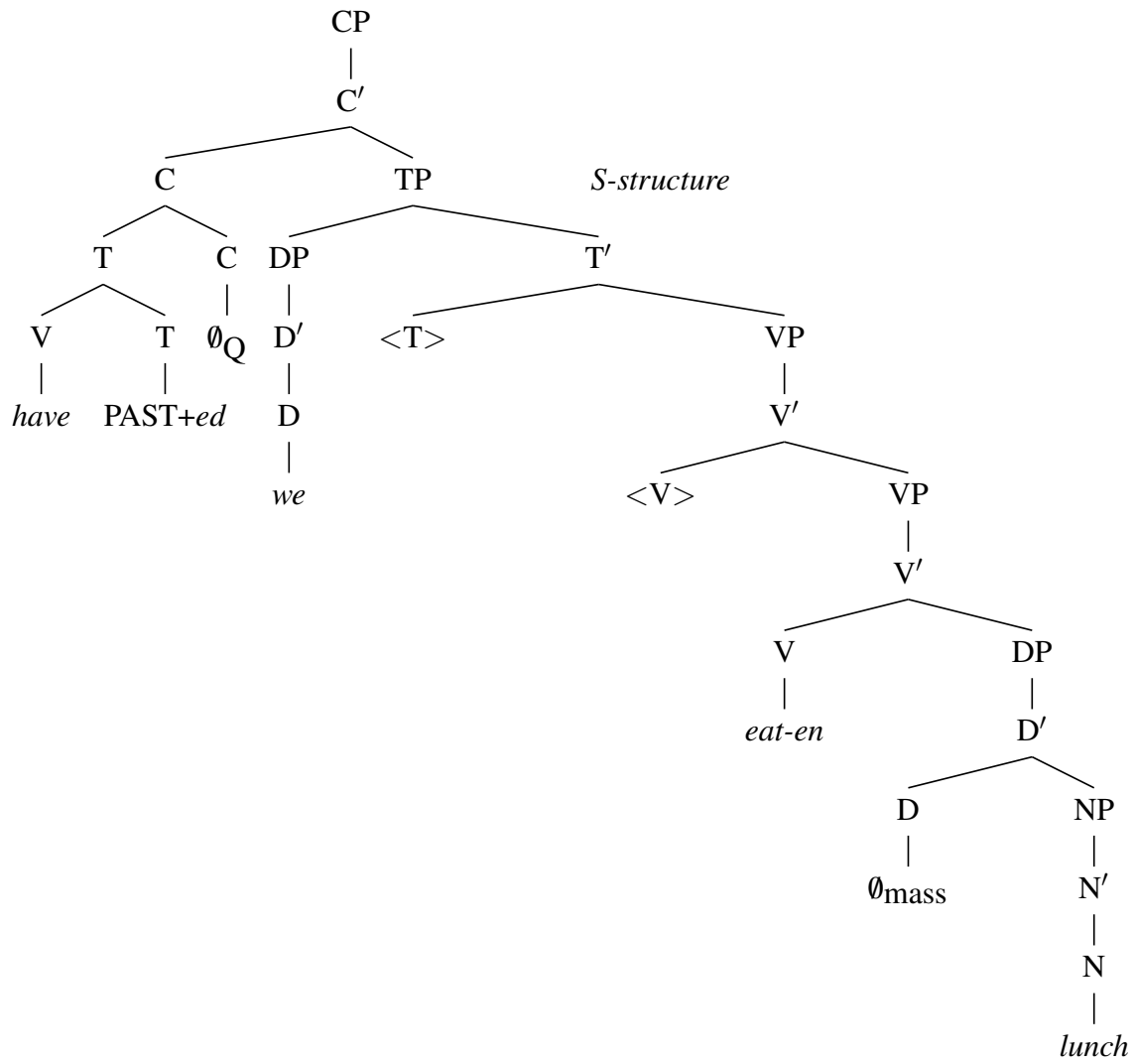
Neg is not an auxiliary, it does not move. Main verbs (like *eat*) are not auxiliaries, they do not move.

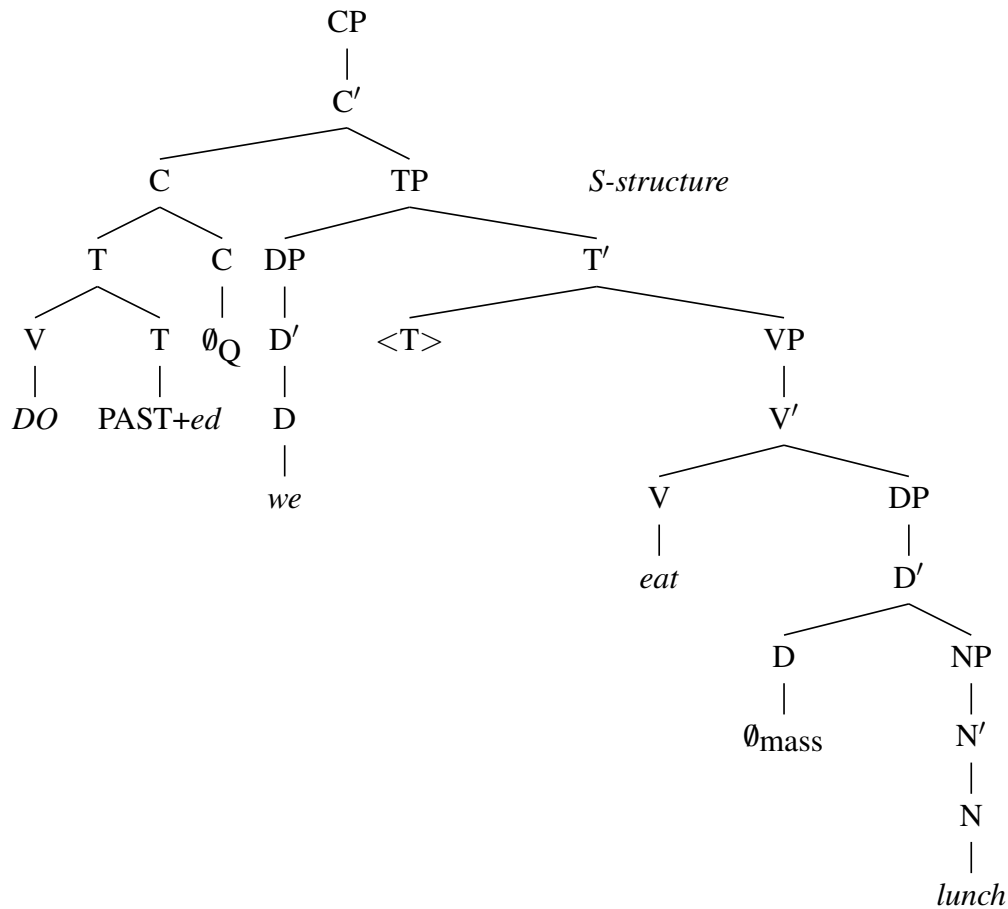
4 Yes-no questions

Having motivated movement to T for auxiliaries, we now have a straightforward way to understand inversion in yes-no questions.

(60) Had we eaten lunch?

Same thing, but instead of saying that [+AUX] V moves to T, we can say that T moves to [+Q] C.





5 Wh-questions

- (61) Lisa thinks that Bart will hug Maggie.
 (62) Lisa wonders whether Bart will hug Maggie.
 (63) * Lisa wonders that Bart will hug Maggie.
 (64) * Lisa thinks whether Bart will hug Maggie.
 (65) * Lisa thinks that Bart will hug.

- (66) *think*, V, [+__ CP_[+Decl]]
 (67) *wonder*, V, [+__ CP_[+Q]]
 (68) *hug*, V, [+__ DP]

- (69) *whether*, C, [+Q]
 (70) *that*, C, [+Decl], [+ __ TP_[+Tense]]

- (71) Lisa wonders who Bart will hug.

- (72) * Lisa thinks who Bart will hug Maggie.
- (73) Lisa asked who Bart will hug at noon at school.
- (74) Lisa asked who will hug Maggie at noon at school.
- (75) Lisa asked when Bart will hug Maggie at school.
- (76) Lisa asked where Bart will hug Maggie at noon.
- (77) Bart will hug Maggie at noon at school.
- (78) Bart will hug Maggie at noon at school.
- (79) Bart will hug Maggie at noon at school.
- (80) Bart will hug Maggie at noon at school.
- (81) Lisa asked who Bart will hug at noon at school.
- (82) Lisa asked who will hug Maggie at noon at school.
- (83) Lisa asked when Bart will hug Maggie at school.
- (84) Lisa asked where Bart will hug Maggie at noon.
- (85) Lisa wonders which child Bart will hug.
- (86) Lisa wonders which child will hug Maggie.

where	there
when	then
what	that
why	hmm
who	ok
how	well

So: *wh*-words are not category C, they can be various categories.

They seem to be doing two jobs: a) the one the pro-form does, b) marking a question.

Question marking is something C is involved with; CP is [+Q] and *wonder* cares.

Being an argument is something V is involved with. V might be [+__ DP].

Another case where we have one thing apparently doing two jobs:

- (87) Bart wants Lisa to hug Maggie.
- (88) Bart wants to hug Maggie.
- (89) Bart persuaded Lisa to hug Maggie.

So maybe something similar is happening here.

- (90) Lisa asked who Bart will hug [who] at noon at school.
- (91) Lisa asked who [who] will hug Maggie at noon at school.
- (92) Lisa asked when Bart will hug Maggie [when] at school.
- (93) Lisa asked where Bart will hug Maggie at noon [where].

The loud *wh*-word goes with the [+Q] CP, and the quiet one is down in the VP.

- (94) She expects [Lisa to embarrass herself].
- (95) Who does she expect [[who] to embarrass herself]?

- (96) Lisa/Homer think that Bart will prefer this picture of himself.
- (97) Which picture of himself does Lisa think that Bart will prefer?
- (98) Which picture of himself does Lisa think that Bart will prefer [wh px of self]?