

## Lecture notes: Week 7

### Auxiliaries

#### 1 Types of auxiliaries

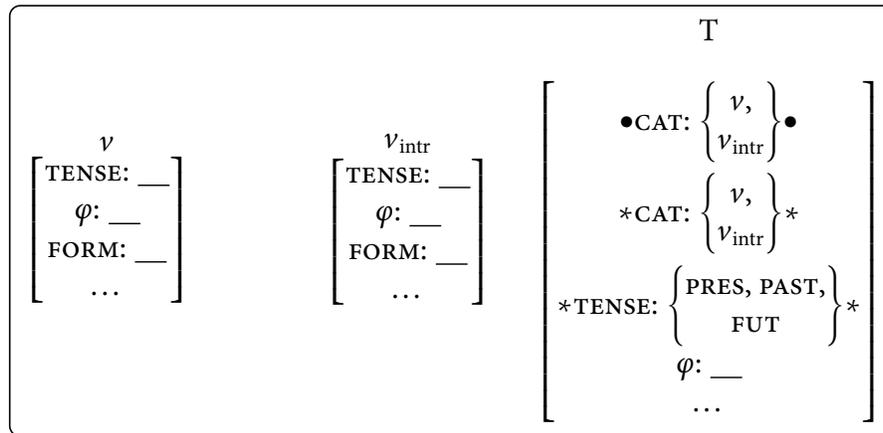
- So far, we have proposed different kinds of functional heads in our grammar. They are given in our (functional) lexicon below.

#### Functional Lexicon

$\left[ \begin{array}{c} \nu \\ \text{CAT: } \nu \\ \bullet \text{CAT: V} \bullet \\ \text{*CAT: V*} \\ \text{TENSE: } \_ \\ \varphi: \_ \\ \text{FORM: } \_ \\ \text{*CASE: ACC*} \\ \bullet \text{N} \bullet \end{array} \right]$	$\left[ \begin{array}{c} \nu_{\text{intr}} \\ \text{CAT: } \nu_{\text{intr}} \\ \bullet \text{CAT: V} \bullet \\ \text{*CAT: V*} \\ \text{TENSE: } \_ \\ \varphi: \_ \\ \text{FORM: } \_ \end{array} \right]$	$\left[ \begin{array}{c} \text{T} \\ \text{CAT: T} \\ \text{*CASE: NOM*} \\ \varphi: \_ \\ \bullet \text{CAT: } \left\{ \begin{array}{c} \nu, \\ \nu_{\text{intr}} \end{array} \right\} \bullet \\ \text{*CAT: } \left\{ \begin{array}{c} \nu, \\ \nu_{\text{intr}} \end{array} \right\} \text{*} \\ \text{*TENSE: } \left\{ \begin{array}{c} \text{PRES, PAST,} \\ \text{FUT} \end{array} \right\} \text{*} \\ \bullet \text{N} \bullet \end{array} \right]$
---	---	---

- We have two  $\nu$  heads, with differing functions:  $\nu$  introduces an agent and assigns accusative case to the object, whereas  $\nu_{\text{intr}}$ , which occurs in passive and unaccusative sentences, does neither of those two things. They both select a VP and agree with its head (triggering head movement) and have inflecting potential both in terms of features from T (i.e. tense and  $\varphi$ ) as well as the possibility to occur in a participle form, e.g. the participle form, by virtue of the feature [FORM:  $\_$ ].
- The T head, on the other hand, has a number of functions. It assigns nominative case to the closest NP that it c-commands and picks up the  $\varphi$ -features of that NP. It can select either  $\nu$  or  $\nu_{\text{intr}}$  and agrees with that head (also passing on tense and  $\varphi$ -features to it, but not triggering head movement). It also requires a nominal in its specifier which, if not the expletive *there*, will be the NP that it assigned nominative case to.
- The entries above are the full lexical specifications which show the full range of features on each kind of head. As you can see, there is quite a lot. Not all of them are relevant for our present discussion so we can omit all those but the features relevant for determining the form of the head in question. The simplified lexicon looks as follows:

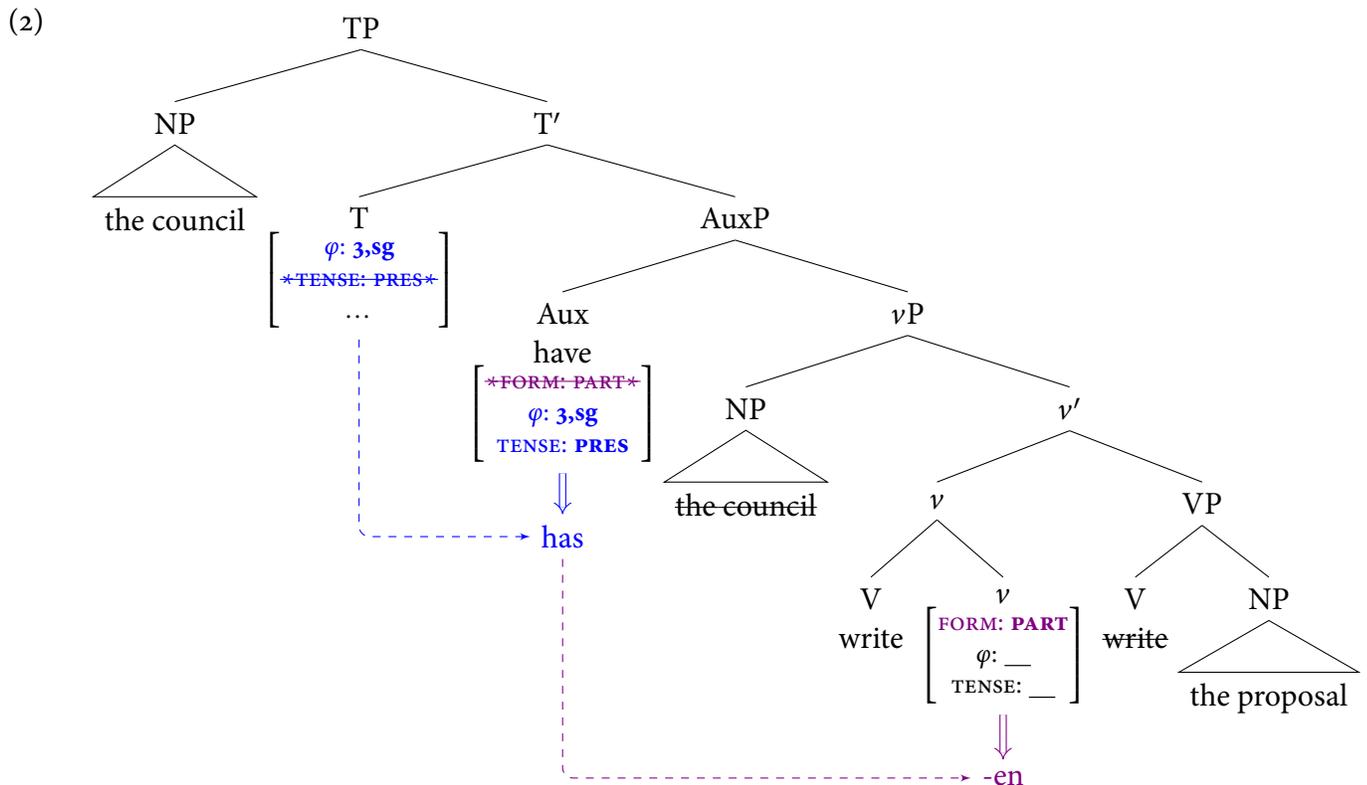
**Functional Lexicon (simplified)**



- In addition to this, we have seen that a sentence can contain auxiliaries:

(1) The council wrote the proposal today  
The council has written the proposal today

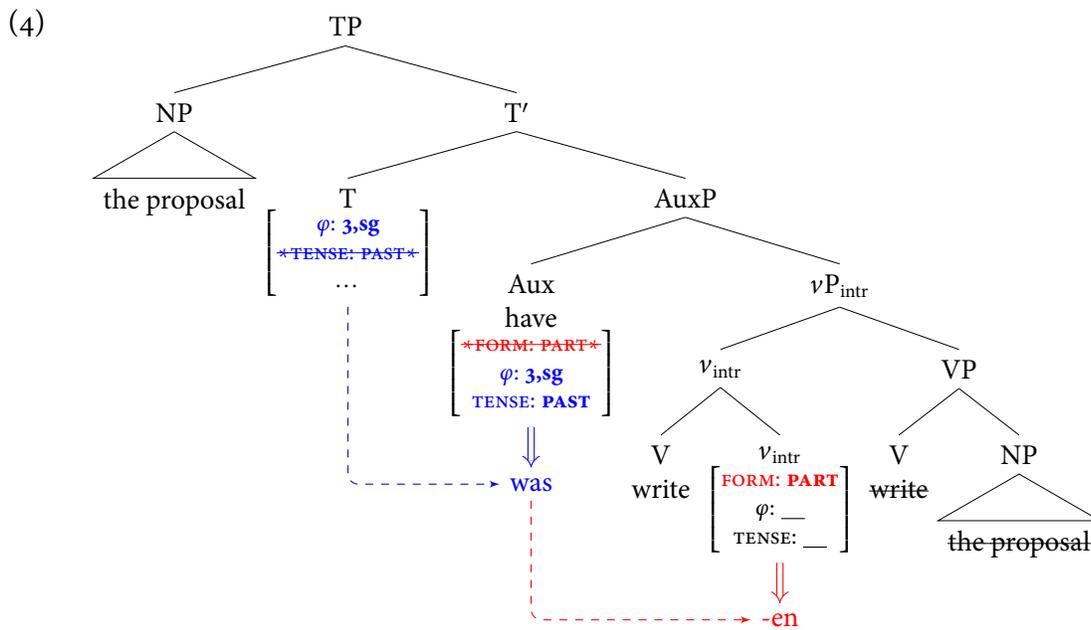
- In the absence of an auxiliary, the tense and  $\varphi$  information from T is passed to  $v$ , leading to *wrote*. In the derivation of *had written*, the past tense feature and 3rd singular value is given to Aux, since it is the closest c-commanded head to T. In turn,  $v$  receives a value for its [FORM] feature from Aux, which causes it to be realized as the participle suffix *-en*.



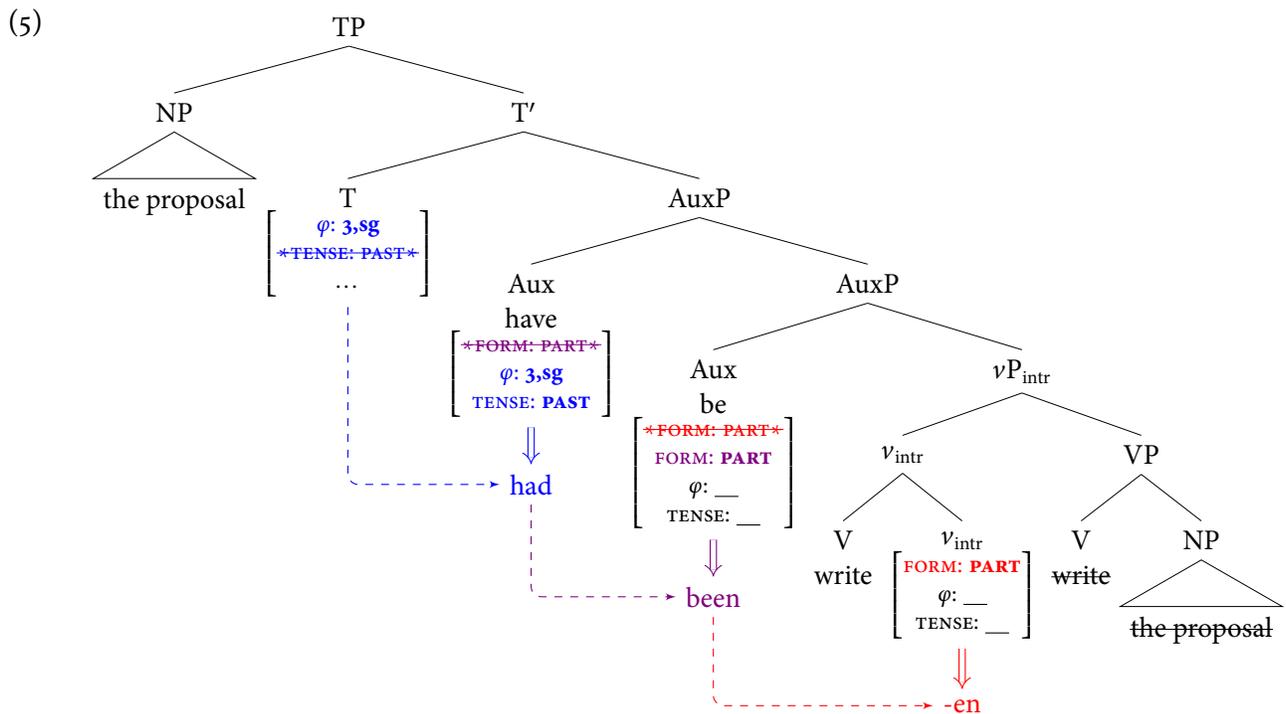
- We have also seen examples of the passive:

(3)

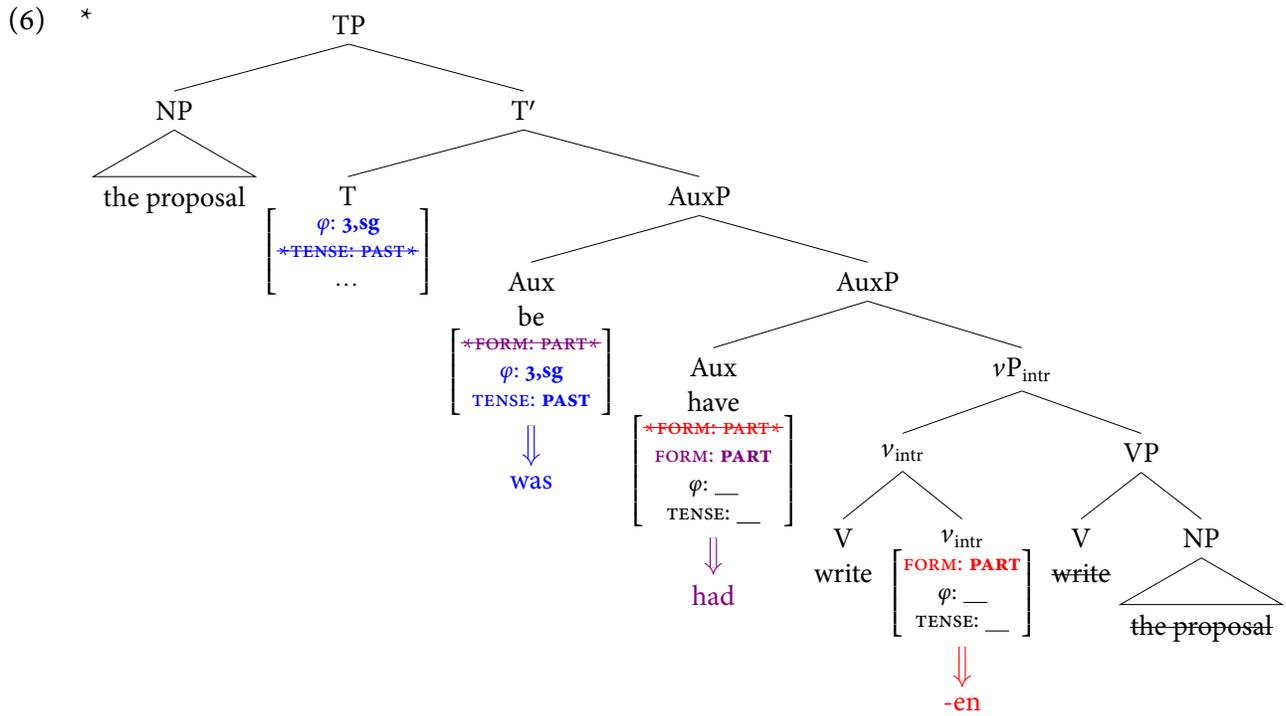
The proposal	was	written	in 2005
The proposal	had	been	written in 2005



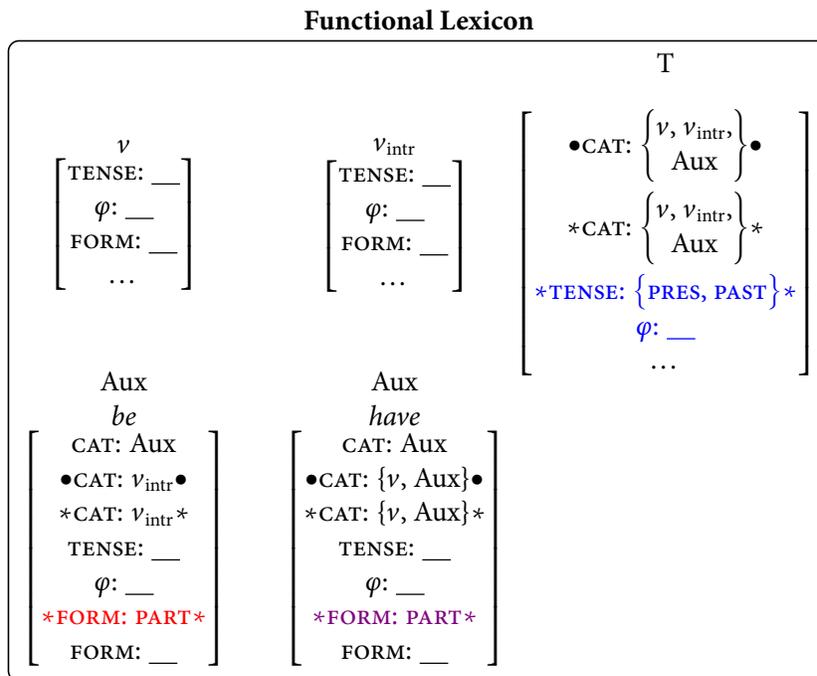
- Can both auxiliaries co-occur? Yes, the following structure shows that:



- What about the reverse order? Can  $Aux_{have}$  select  $Aux_{be}$ ? The following structure shows it is not possible:

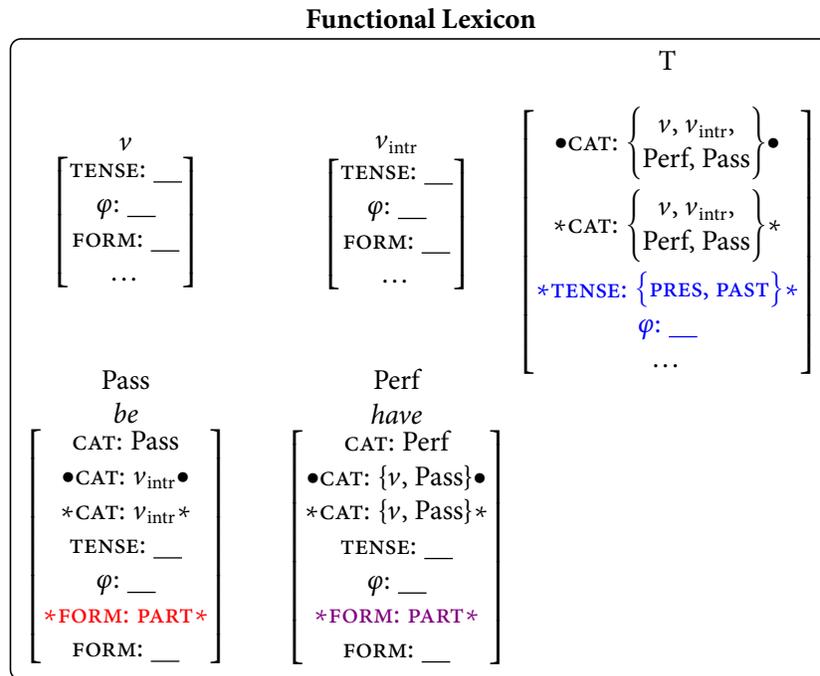


- We can add these two auxiliaries to our lexicon now:

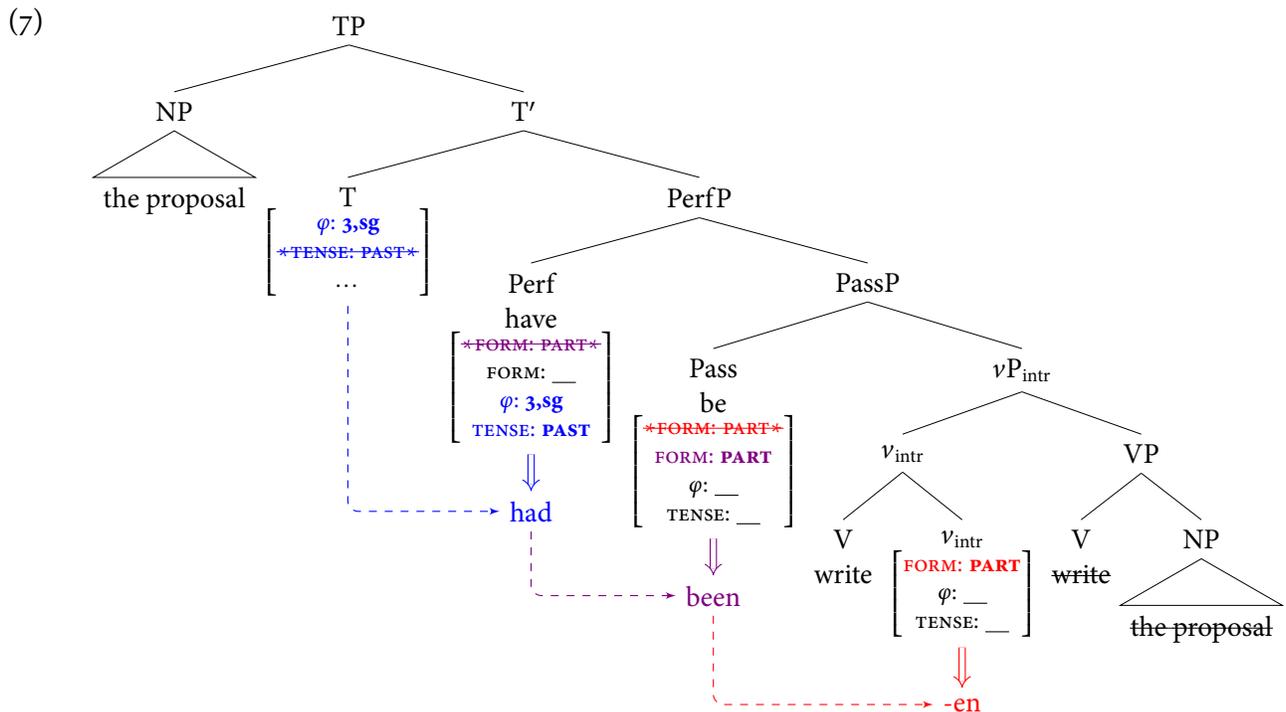


- We also have to update our selectional specifications for each so that Perf can select either  $vP$  or PassP and T can select either auxiliary type.

- It will be useful to be able to distinguish these two different types of auxiliaries. To do this, let's change our assumptions and sub-divide the category Aux into Perf and Pass for the *have* and *be* auxiliary, respectively:



- With these changes in mind, the double auxiliary example in (5) would be analyzed as follows:

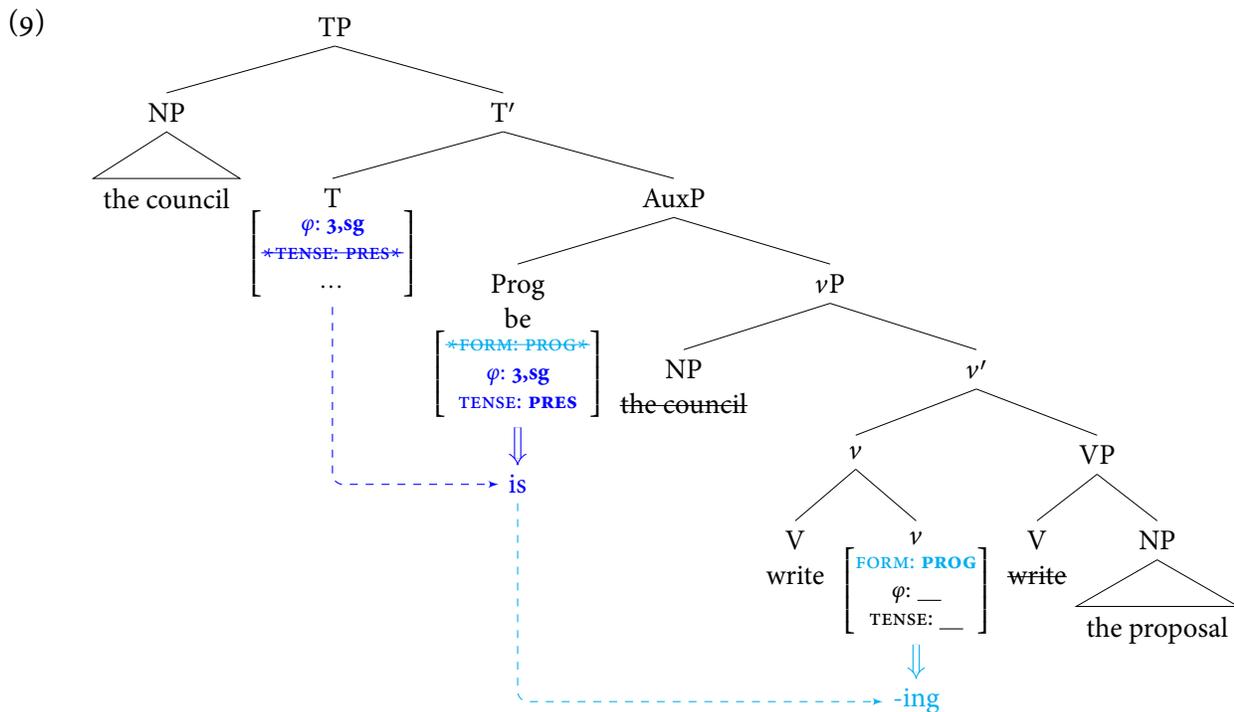


- There is another kind of auxiliary we have not yet talked about yet. This is in a sentence like *The council is writing the proposal today* in (8).

(8)

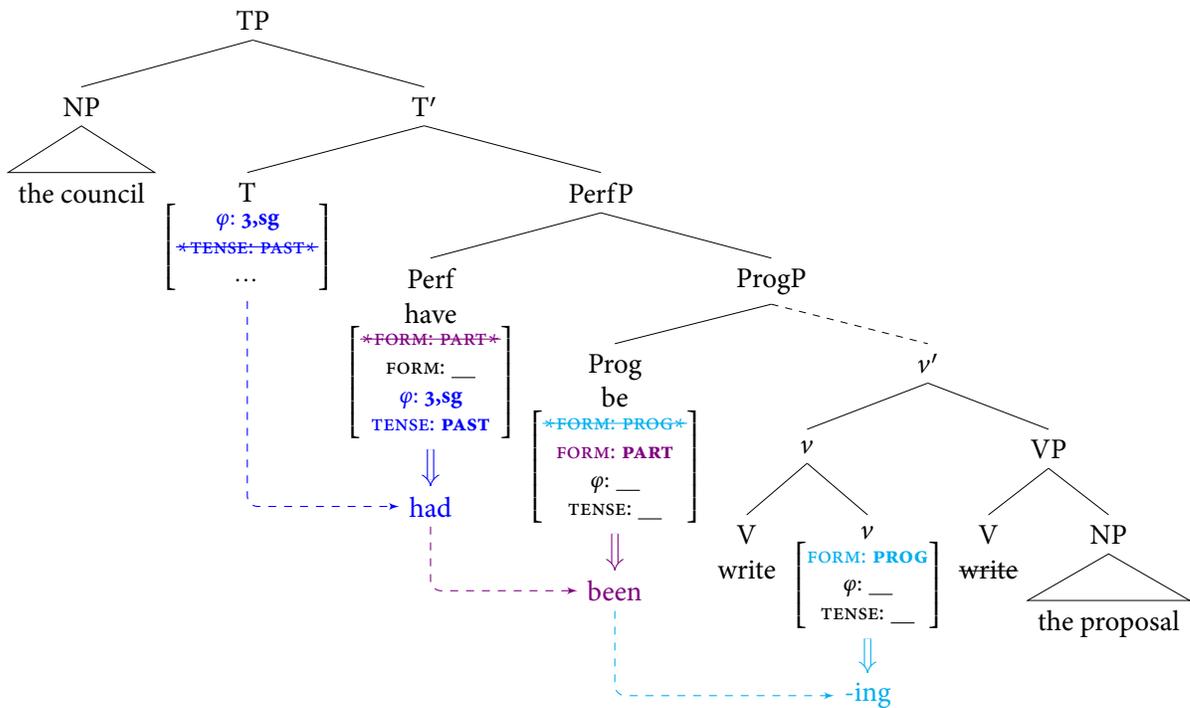
	Perf	Prog	$v+V$		
The council			wrote	the proposal	today
The council	has		written	the proposal	today
The council		is	writing	the proposal	today
The council	has	been	writing	the proposal	since 2005

- Is this the same *be*? It cannot be because, on the one hand, it does not trigger the past participle form of the verb, but rather the continuous/progressive participle ending in *-ing*. Also, we said that the passive auxiliary *be* must select an intransitive  $vP$  and this auxiliary appears to be selecting a transitive  $vP$ .
- For this reason, it seems appropriate to treat it as a different category of auxiliary. We will call it Prog.



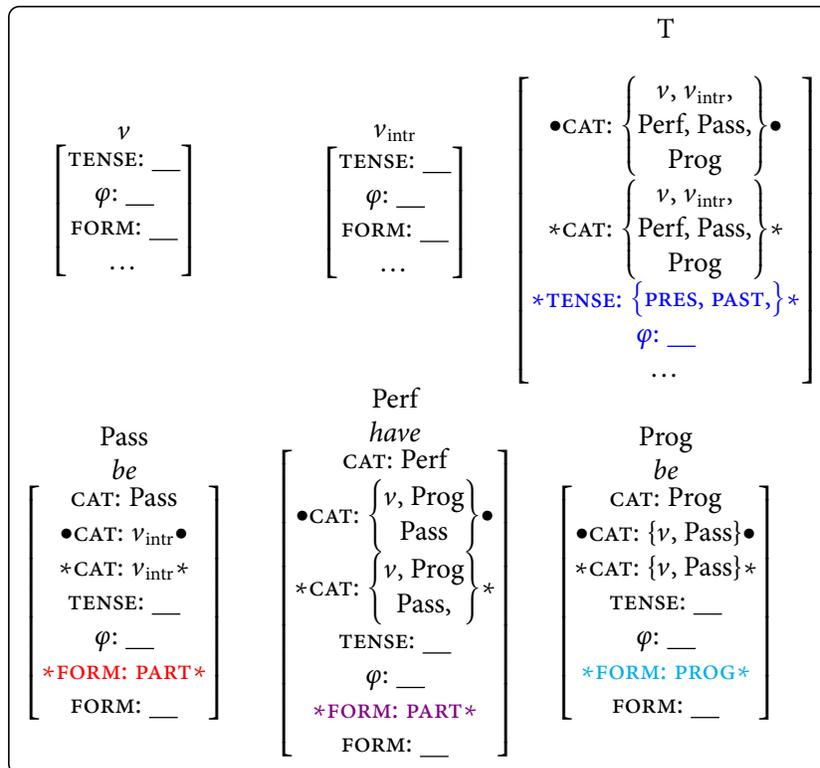
- The important different here is that Prog bears a different value for its form feature, namely [FORM: PROG]. This is the feature that will cause  $v$  to be realized as *-ing* participle form rather than *-ed/-en*.
- This head can co-occur with perfect auxiliary as the sentence *The council has been discussing the proposal* in (8) shows.
- On this analysis, Perf may also select ProgP. Since Perf now agrees with Prog, it will condition the participle form of the progressive auxiliary *be*, namely *been*:

(10)



- Let's add this auxiliary to our lexicon:

Functional Lexicon

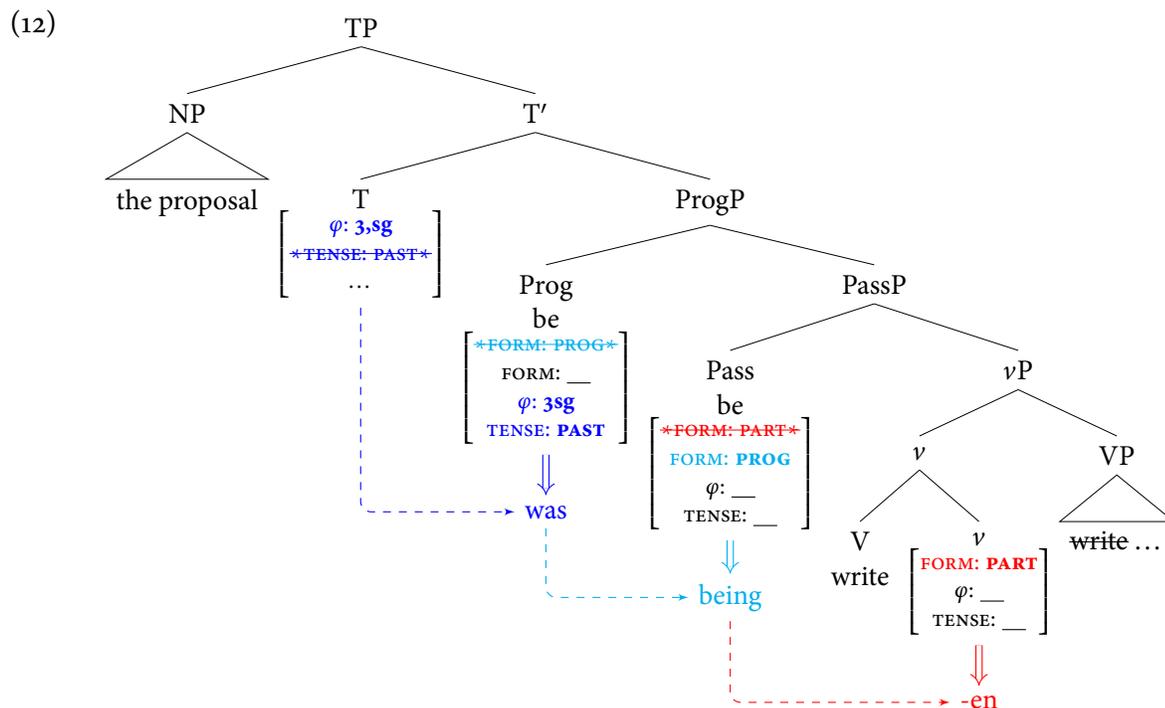


- Now what about the passive auxiliary? Can Prog select this? Yes, the sentence *The proposal was being*

written in 2005 in (11) shows this.

	Perf	Prog	Pass	$v+V$	
The proposal			was	written	in 2005
The proposal	had		been	written	in 2005
The proposal		was	being	written	in 2005
The proposal	had	been	being	written	until 2005

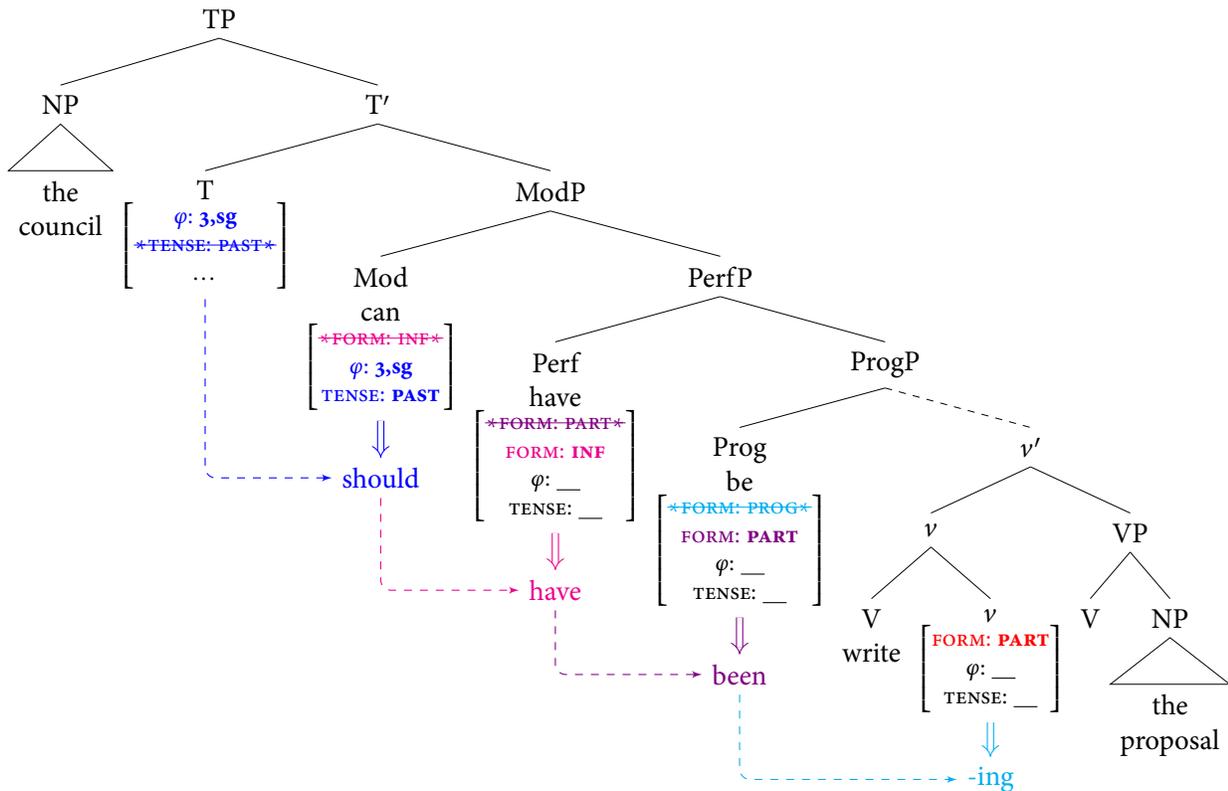
- This can be analyzed as follows:



- Furthermore, it is possible for all three auxiliaries to occur together, as in the sentence *The proposal had been being written until 2005*.
- The analysis predicts the right outcome here. Since Perf may select Prog, it will now be the closest c-commanded head to T and receive the tense and  $\phi$ -features. Furthermore, it will assign its own form specification, namely PART to the progressive auxiliary *be*, causing it surface as *been*.



(15)



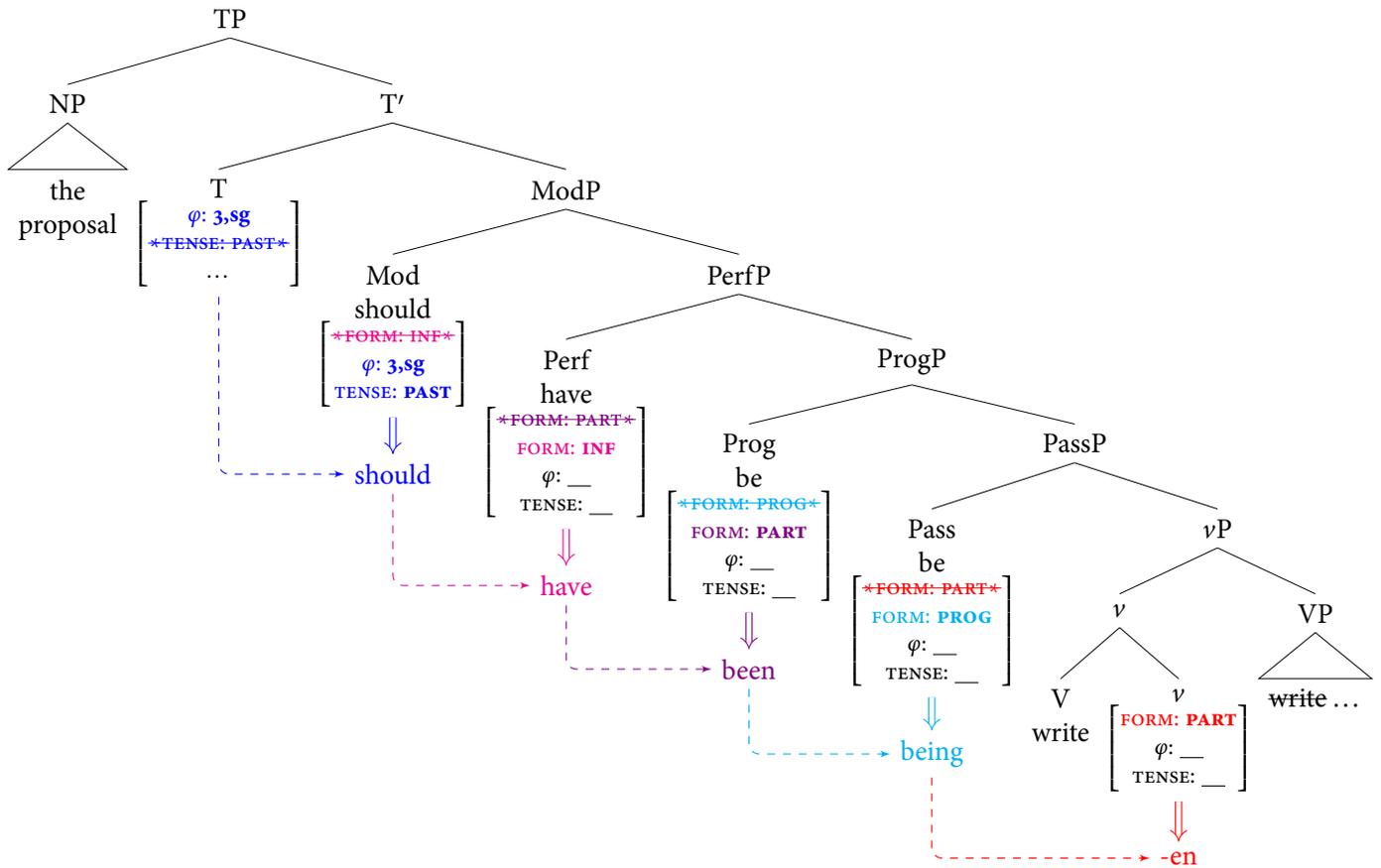
- Furthermore, we can have modals in addition to the other three auxiliaries in passive sentences:

(16)

	Perf	Prog	Pass	$v+V$		
The proposal			was	written	today	
The proposal	had		been	written	today	
The proposal		was	being	written	today	
The proposal	had	been	being	written	today	
The proposal	should	have	been	being	written	today

- The analysis of this more complex sentence is given in (17).
- As expected, the perfect auxiliary surfaces in its infinitive form *have*.

(17)

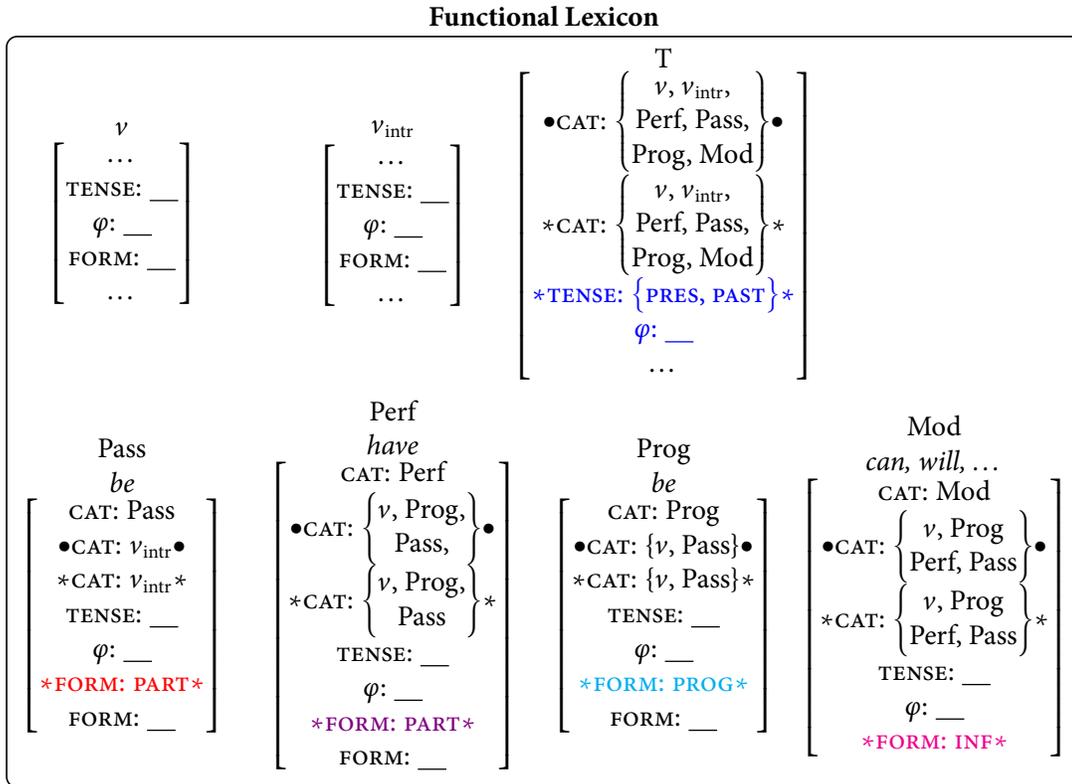


- We can omit any of the auxiliaries in this tree (except for Pass) and the form of the remaining auxiliaries is determined by whatever the closest c-commanding head is (or linearly preceding in the table below):

(18)

	Modal	Perf	Prog	Pass	$v+V$	
The proposal				was	written	today
The proposal		had		been	written	today
The proposal			was	being	written	today
The proposal		had	been	being	written	today
The proposal	should	have	been	being	written	today
The proposal	should		be	being	written	today
The proposal	should	have		been	written	today
The proposal	should			be	written	today

- With all of this in place, we have run out of auxiliaries to add. Let's add Mod to our functional lexicon:



- As we can see, there is a single entry for each auxiliary regardless of their form or function in the sentence. The way the system is set up (with Agree to check and value features) will mean that the right combination of auxiliaries will just fall out of whatever auxiliaries we try to combine from the functional lexicon.

## 2 Negation

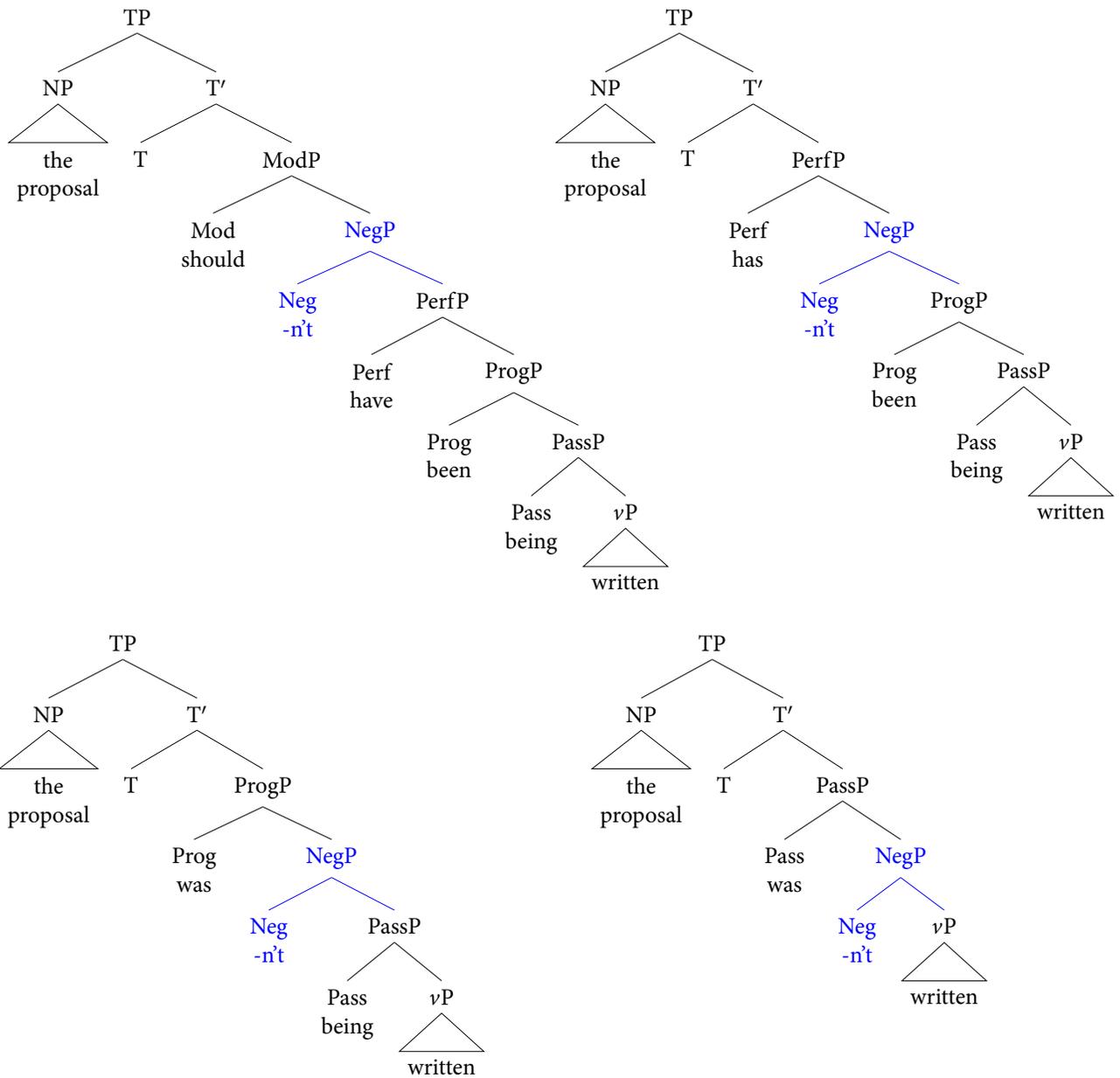
- While we have exhausted the different auxiliaries we can add to our sentences, there is another element that we have not talked about, namely negation.
- Let's consider the position in which the so-called 'affixal' negation *-n't* may surface

(19)

	Modal	Perf	Prog	Pass	v+V	
The proposal				was( <b>n't</b> )	written( <b>*n't</b> )	yet
The proposal		had( <b>n't</b> )		been( <b>*n't</b> )	written( <b>*n't</b> )	yet
The proposal			was( <b>n't</b> )	being( <b>*n't</b> )	written( <b>*n't</b> )	yet
The proposal		had( <b>n't</b> )	been( <b>*n't</b> )	being( <b>*n't</b> )	written( <b>*n't</b> )	yet
The proposal	should( <b>n't</b> )	have( <b>*n't</b> )	been( <b>*n't</b> )	being( <b>*n't</b> )	written( <b>*n't</b> )	yet

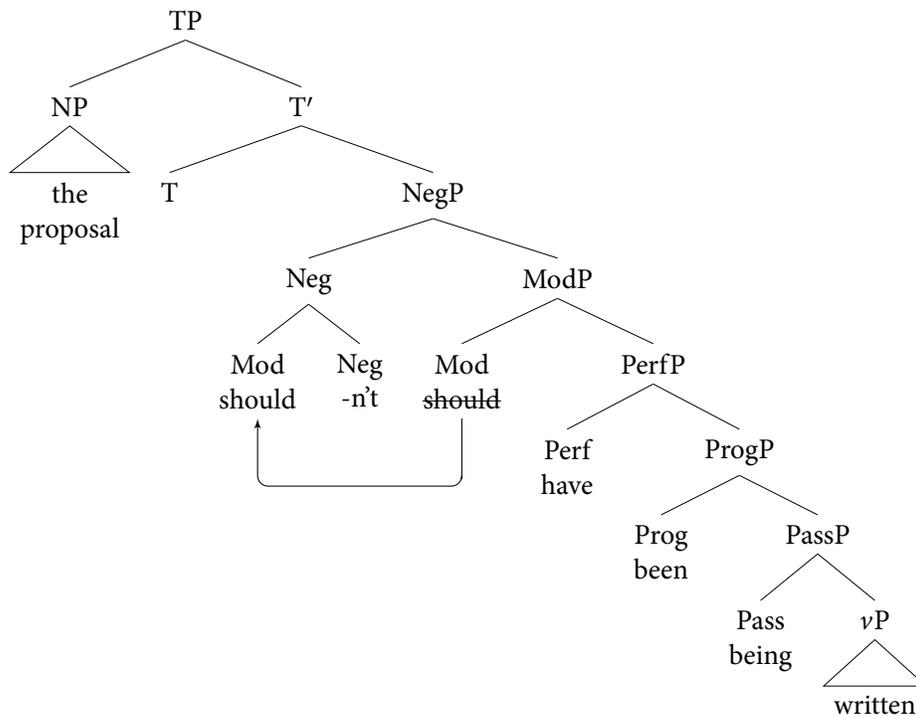
- This negation may only show up after the first auxiliary in the sentence, but this can be any of those.
- At first glance, this might suggest that NegP can select any of the following projections: PerfP, ProgP, PassP, vP, as shown by the trees below.

(20)



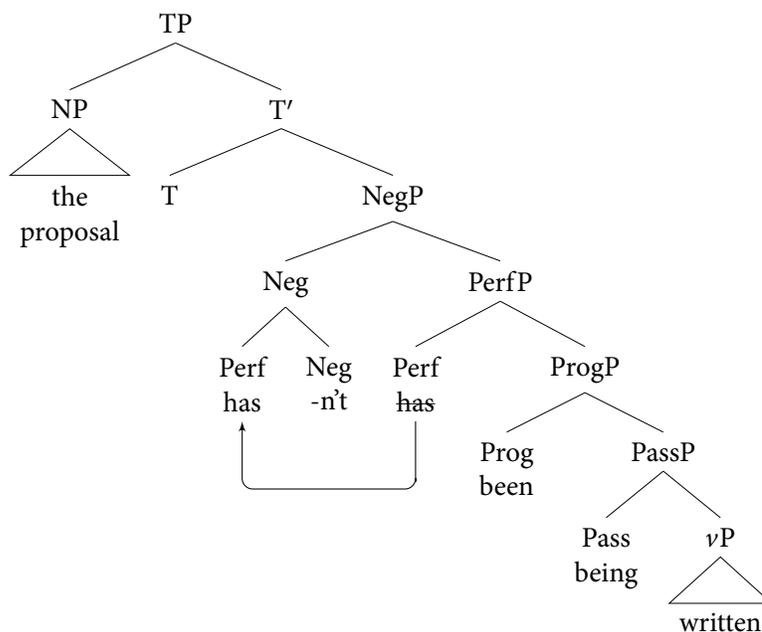
- Of course, the problem with this approach is that it predicts that all of these positions for *-n't* should be available all the time and this is clearly wrong. The correct generalization seems to be that *-n't* may appear after the first/highest auxiliary in the clause, regardless of what that auxiliary is.
- This is a generalization that we can capture elegantly if we say that NegP actually has a fixed position in the clause above all of the auxiliaries and triggers head movement of the closest c-commanded auxiliary. So, the modal will move whenever it is present:

(21)



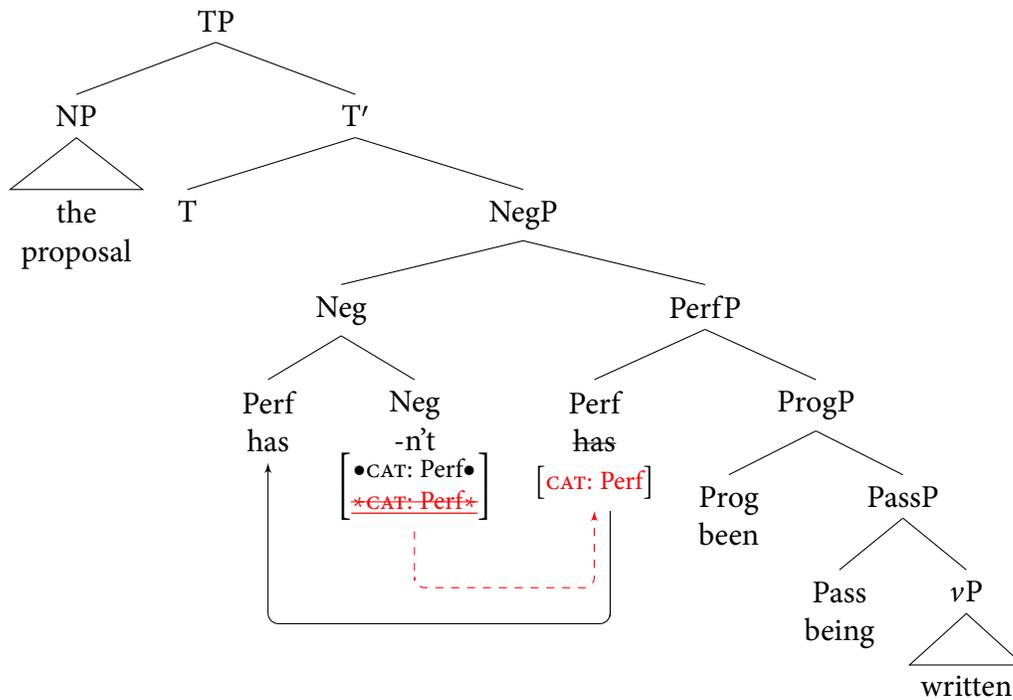
- In the absence of the modal, then Perf will move to Neg if it is the next highest auxiliary, and so on and so forth:

(22)



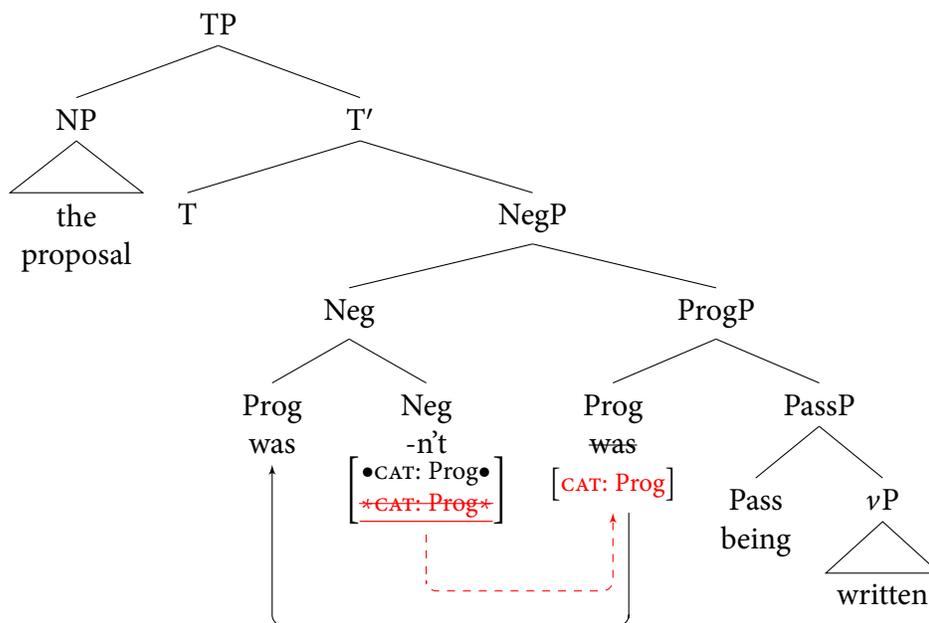
- We already know how to trigger head movement; we can give Neg a strong \*-category feature so that it will trigger head movement of the head of whatever projection it selects. I will show this below the example above:

(23)



- Given our assumption that head movement triggering features are always linked to selection, then only the closest auxiliary can be moved. If we omit the PerfP in this example, then Prog will undergo head movement to Neg instead:

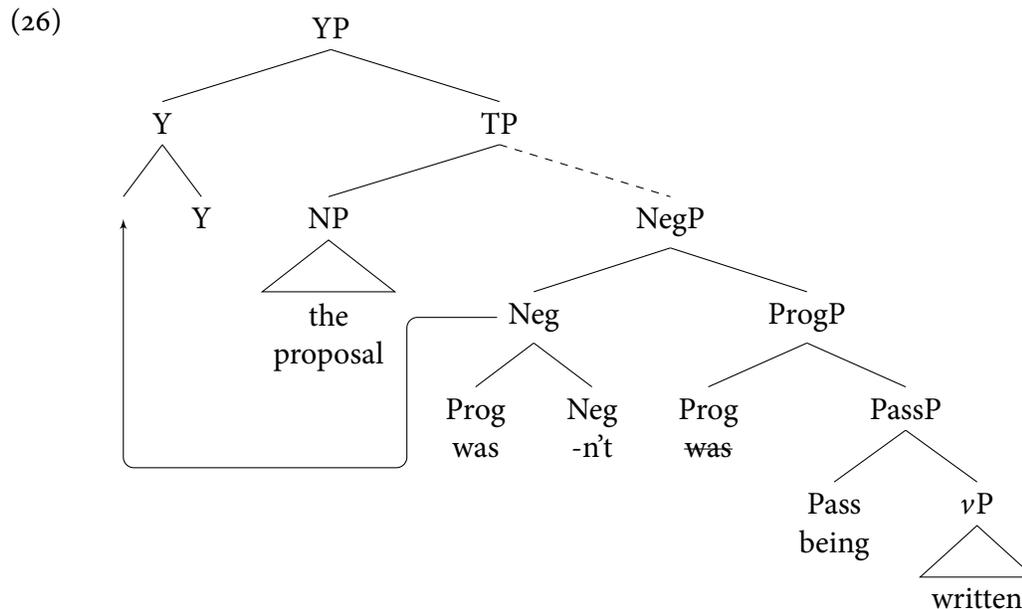
(24)



- On this analysis, the negative marker *-n't* and the auxiliary now form an indivisible unit. This is a good thing since when the auxiliary moves to a head position higher than TP in questions for example, the negation is taken along with it:

- (25) a. Was the proposal being written?  
b. Wasn't the proposal being written?

- We will come back to this exact position Y, but this is clearly a good result of our assumption that the two form a complex head, since they undergo head movement together:



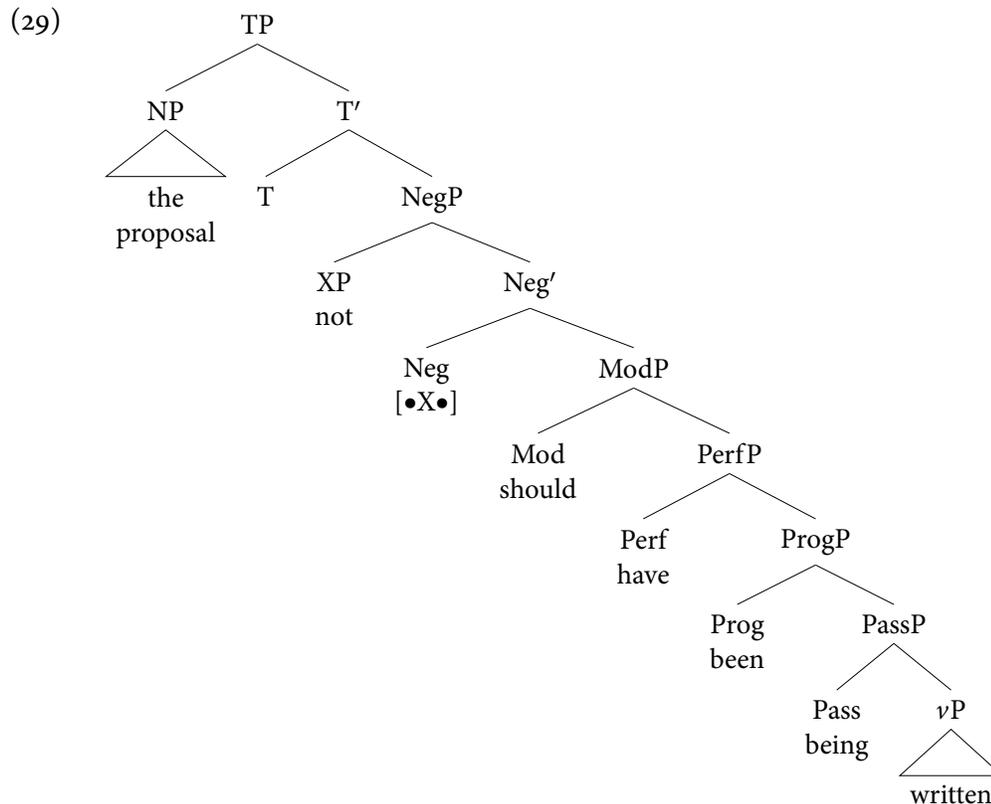
- With this in mind, let's compare this with the negator *not*. We could give this the same analysis as *-n't* in which it is the head of a NegP. But this would incorrectly predict that it forms the same kind of tight unit with the auxiliary (a complex head). But as we can clearly see, it does not invert with respect to the subject like *-n't* does:

- (27) a. The proposal was not being written.  
b. \*Was not the proposal being written?

- It seems preferable to treat *not* as a phrase rather than a head. A natural choice would be to assume it is an adjunct, but we already argued against this possibility due to the fact that negation seems to block assignment of tense/ $\phi$ -features to  $v$  whereas adjuncts do not, leading to insertion of *do*.

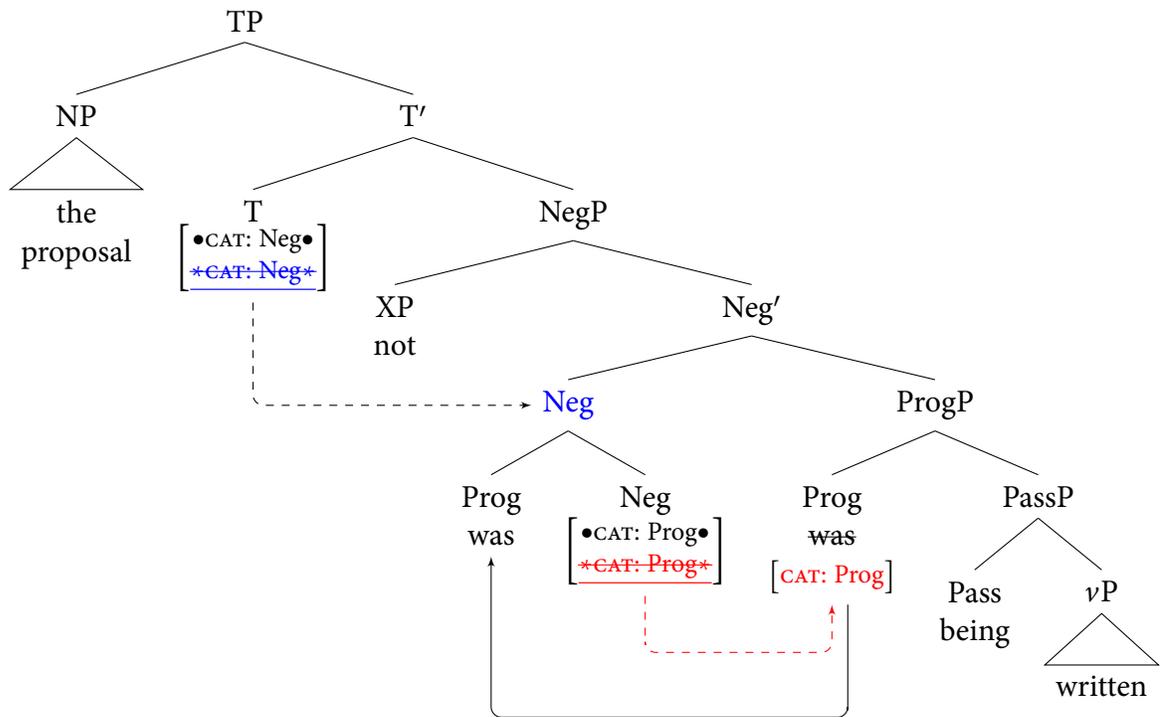
- (28) a. The council finally wrote the proposal.  
b. \*The council not wrote the proposal.  
c. The council did not write the proposal.

- For this reason, it seems preferable to treat *not* as involving a NegP projection, just like with *-n't*. If it isn't the head of NegP, then where could it be? Let's assume that *not* is actually a 'negative particle', i.e. a phrase. Just like other particles (e.g. those associated with verbs), we can treat it as a phrasal projection of category X. There will therefore be another variant of Neg whose head has no overt realization, but selects *not* as its specifier:

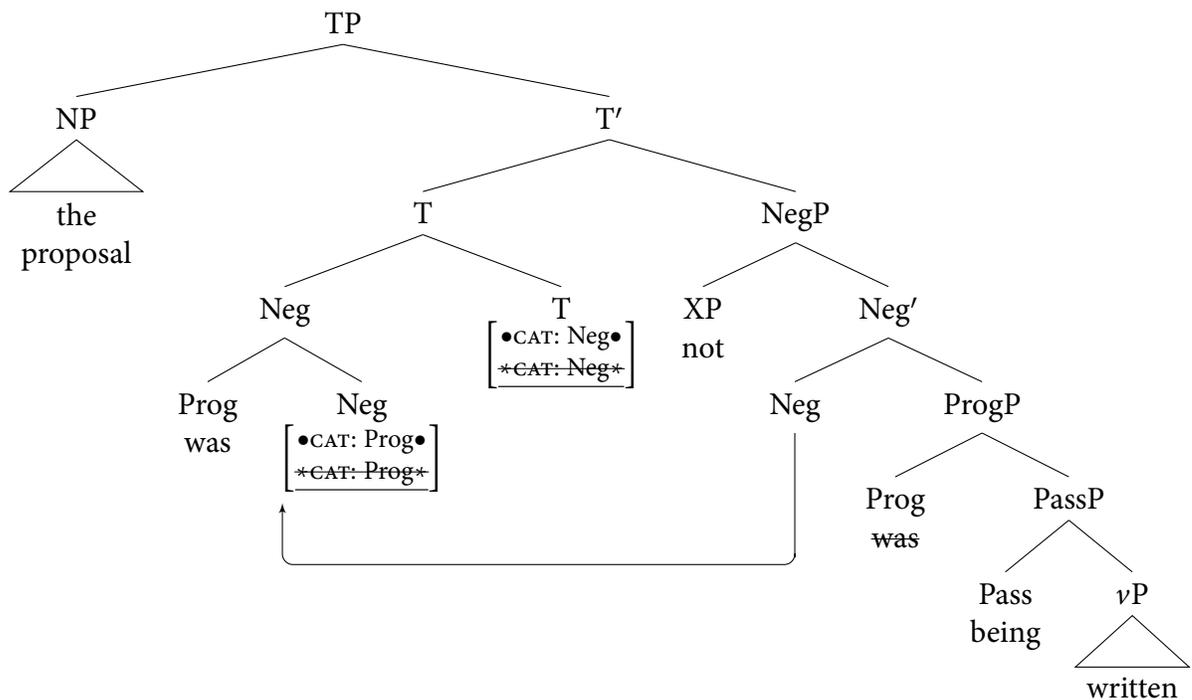


- The same generalization we had for *-n't* also applies here: The highest auxiliary in the clause must precede *not*. For this reason, we would ideally want to Mod to move to T in (29). However, our mechanism for head movement only allows movement of the closest c-commanded head (it is not possible to skip heads). Since T selects NegP in (29), the strong feature it would have is [\*CAT: Neg\*], but not [\*CAT: Mod\*].
- We can solve this issue, however, if say that Neg still triggers head movement the closest c-commanded auxiliary (i.e. bears a strong Agree feature) even when it is null. On this view, the auxiliary will first move to null Neg and then Neg+Aux will move to T where it will precede the phrase *not* in the specifier of Neg.
- This will look as follows for sentence *The proposal was not being written*:

(30)



(31)



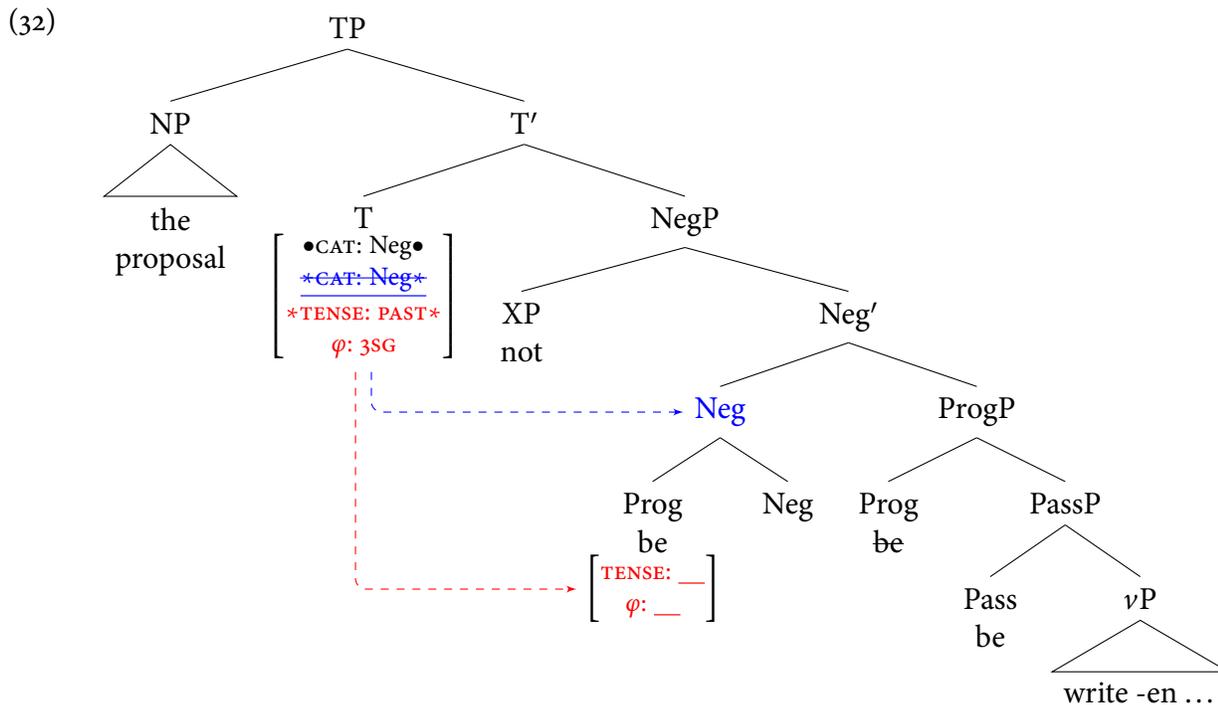
- The final thing we need need to do then is to add the two Neg heads to our functional lexicon:

## Functional Lexicon

$\nu$ $\left[ \begin{array}{c} \dots \\ \text{TENSE: } \_ \\ \varphi: \_ \\ \text{FORM: } \_ \\ \dots \end{array} \right]$	$\nu_{\text{intr}}$ $\left[ \begin{array}{c} \dots \\ \text{TENSE: } \_ \\ \varphi: \_ \\ \text{FORM: } \_ \\ \dots \end{array} \right]$	$T$ $\left[ \begin{array}{c} \bullet \text{CAT: } \left\{ \begin{array}{c} \nu, \nu_{\text{intr}}, \text{Perf, Pass,} \\ \text{Prog, Mod, Neg} \end{array} \right\} \bullet \\ * \text{CAT: } \left\{ \begin{array}{c} \nu, \nu_{\text{intr}}, \text{Perf, Pass,} \\ \text{Prog, Mod, Neg} \end{array} \right\} * \\ * \text{TENSE: } \{ \text{PRES, PAST} \} * \\ \varphi: \_ \\ \dots \end{array} \right]$	
Pass <i>be</i> $\left[ \begin{array}{c} \text{CAT: Pass} \\ \bullet \text{CAT: } \nu_{\text{intr}} \bullet \\ * \text{CAT: } \nu_{\text{intr}} * \\ \text{TENSE: } \_ \\ \varphi: \_ \\ * \text{FORM: PART} * \\ \text{FORM: } \_ \end{array} \right]$	Perf <i>have</i> CAT: Perf $\left[ \begin{array}{c} \bullet \text{CAT: } \left\{ \begin{array}{c} \nu, \text{Prog,} \\ \text{Pass,} \end{array} \right\} \bullet \\ * \text{CAT: } \left\{ \begin{array}{c} \nu, \text{Prog,} \\ \text{Pass} \end{array} \right\} * \\ \text{TENSE: } \_ \\ \varphi: \_ \\ * \text{FORM: PART} * \\ \text{FORM: } \_ \end{array} \right]$	Prog <i>be</i> CAT: Prog $\left[ \begin{array}{c} \bullet \text{CAT: } \{ \nu, \text{Pass} \} \bullet \\ * \text{CAT: } \{ \nu, \text{Pass} \} * \\ \text{TENSE: } \_ \\ \varphi: \_ \\ * \text{FORM: PROG} * \\ \text{FORM: } \_ \end{array} \right]$	Mod <i>can, will, ...</i> CAT: Mod $\left[ \begin{array}{c} \bullet \text{CAT: } \left\{ \begin{array}{c} \nu, \text{Prog} \\ \text{Perf, Pass} \end{array} \right\} \bullet \\ * \text{CAT: } \left\{ \begin{array}{c} \nu, \text{Prog} \\ \text{Perf, Pass} \end{array} \right\} * \\ \text{TENSE: } \_ \\ \varphi: \_ \\ * \text{FORM: INF} * \end{array} \right]$
Neg <i>-n't</i> CAT: Neg $\left[ \begin{array}{c} \bullet \text{CAT: } \left\{ \begin{array}{c} \nu, \nu_{\text{intr}}, \\ \text{Perf, Pass,} \\ \text{Prog, Mod} \end{array} \right\} \bullet \\ * \text{CAT: } \left\{ \begin{array}{c} \nu, \nu_{\text{intr}}, \\ \text{Perf, Pass,} \\ \text{Prog, Mod} \end{array} \right\} * \end{array} \right]$	Neg CAT: Neg $\left[ \begin{array}{c} \bullet \text{CAT: } \left\{ \begin{array}{c} \nu, \nu_{\text{intr}}, \\ \text{Perf, Pass,} \\ \text{Prog, Mod} \end{array} \right\} \bullet \\ * \text{CAT: } \left\{ \begin{array}{c} \nu, \nu_{\text{intr}}, \\ \text{Perf, Pass,} \\ \text{Prog, Mod} \end{array} \right\} * \\ \bullet \text{CAT: X} \bullet \end{array} \right]$		

- A couple of things are worth noting here: We given both Neg heads a strong  $*$ -feature for auxiliaries, but crucially not for  $\nu$  or  $\nu_{\text{intr}}$ . The same is true for T, it will attract any auxiliary or Neg head that is the head of the phrase it selects. Again, we don't want T to trigger head movement of verbs, so the  $\nu$  Agree features are weak.
- There is a minor technical issue here that I have not discussed here. When T selects Neg it will agree that head by virtue of having  $[* \text{Neg} *]$ . But it will need to pass on its tense and  $\varphi$ -feature to whatever the highest auxiliary is below Neg. While we have often talked about checking of these features as a side effect of the category  $*$ -feature on a given head, here it seems we would have to allow  $[* \text{TENSE} *]$  to probe independently to find the highest Aux.

- So, the example we discussed above with *not* would actually look like this:



- In summary, this is full table for the auxiliary possibilities as involving movement across/through the position of negation in a passive sentence:

(33)

	T	Neg	Modal	Perf	Prog	Pass	$\nu$ +V
The proposal	was	-n't/not				—	written today
The proposal	had	-n't/not		—		been	written today
The proposal	was	-n't/not			—	being	written today
The proposal	had	-n't/not		—	been	being	written today
The proposal	should	-n't/not	—	have	been	being	written today