LX 321/521/621 Syntax
Spring 2024

Homework #7
DUE MON APR 8

1 Movement and auxiliaries

Here are some sentences with structures that involve movement. We assume:

- First, the tree is constructed according to the rules and lexicon.
- Then, after that, things that move are re-located.

In particular, there is a "deep structure" that exists before things move, and then things move to result in the structure that ultimately gets pronounced (the "surface structure").

The rules for *do*-support are:

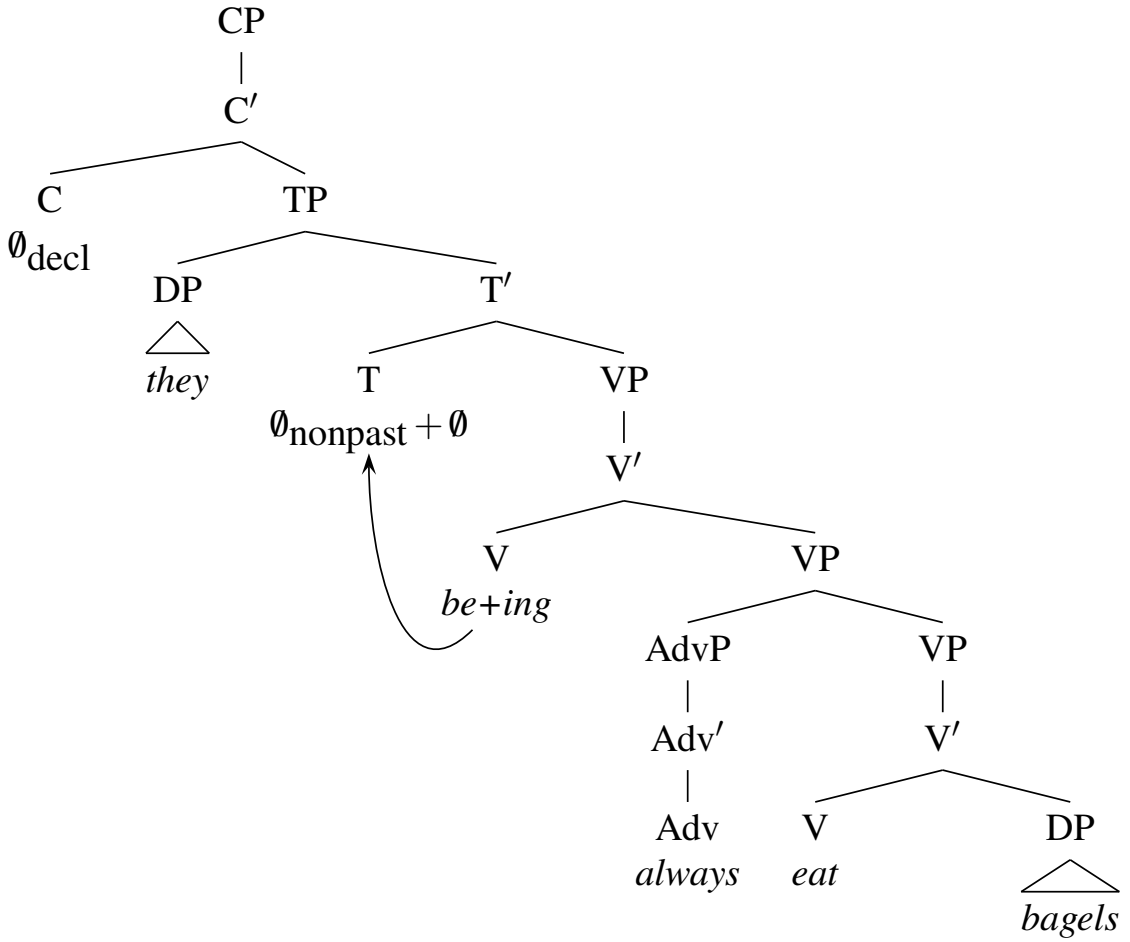
- If: The complement of T (sibling of T) is not VP
- And: V does not move to T (so, V is not [+AUX])
- Then: T ([+PAST] or [–PAST]) gets pronounced as a form of *do* (write *does*/*did*/etc. under T).

When drawing the trees, draw the "deep structure" (pre-movement) tree first, then draw an arrow from the initial position to the ending position of the thing that's moving. (The ending position in this problem will always be T.) All of these trees contain "they" and "bagels"—you can use DP and a triangle for these, as in the example below. The important part here is the stuff between the subject and the object really.

Example. They are always eating bagels.

If you are using `jssyntaxtree`, you can use the following code for this.

```
[CP [C' [C Ødecl ]
[TP [DP they . ] [T' [T Ønonpast+Ø ]
[VP [V' [V be+ing -> 3]
[VP [AdvP [Adv' [Adv always]]]
[VP [V' [V eat] [DP bagels .]]]]]]]]]
```



Note on tree above: I have attached *always* to the lowest VP. That is probably right. But there is nothing that really rules out the possibility of attaching it to the VP headed by *be+ing* instead. I wrote $\emptyset_{\text{nonpast}}$ for T, elsewhere I've written [–PAST] for T. Either is fine. I also wrote $+\emptyset$ for the tense suffix (rather than, say, $+s$) on the assumption that this is already in the regular agreeing form for 3pl (like *they*), even though in this case it leads to an irregular pronunciation *are*. Fundamentally, this is a non-past T node that has no pronunciation of its own, and contributes a non-past tense suffix that agrees with the subject in ϕ-features. If that is expressed accurately, whatever variation you use is fine.

Your task: Draw trees for each of the following sentences.

(1) They have eaten bagels.

(2) They should have eaten bagels.

(3) They should not have eaten bagels.

(4) They should not have been eating bagels.

(5) They are eating bagels.

(6) They did not eat bagels.

(7) They never eat bagels.

2 Yes-no questions

2.1 Main clause questions

Your task: Draw trees for each of the following sentences. In simple yes-no questions, T moves to C. Same rules as above, so draw the tree with everything in its pre-movement position, then indicate where things move with arrows. Whenever there is an auxiliary that moves to T, this means there will be two arrows, one from the auxiliary to T, and another from T to C. You can keep drawing triangles for *they* and *bagels*.

(8) Have they eaten bagels?

(9) Do they always eat bagels?

(10) Should they not eat bagels?

2.2 Embedded question

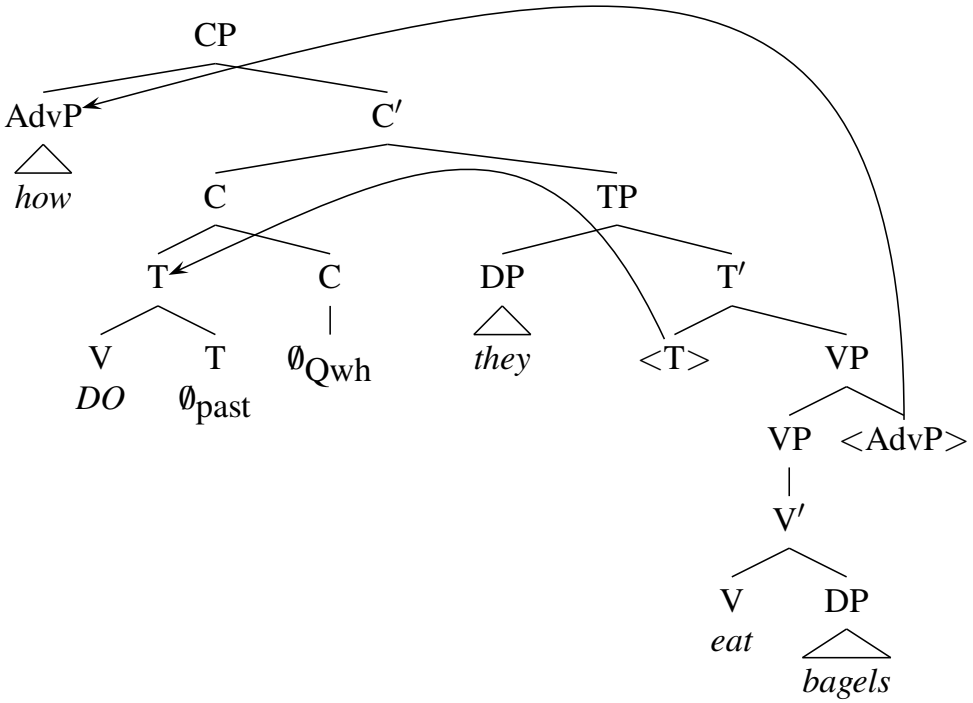
Your task: Draw a tree for the following sentence. You can keep drawing triangles for *they* and *bagels*, and also for *we*.

(11) We wonder if they have not eaten bagels.

3 Wh-questions

In this section, we'll do structures for a few *wh*-questions. This time, new policy for drawing the trees. Instead of drawing the pre-movement position of everything, we will draw the post-movement position. So, where a *wh*-word moves, draw $< >$ around its phrasal node in the original position, an arrow to its final position, and the *wh*-word in the final position. Where V moves to T or T moves to C, draw <T> or <V> in the original position, an arrow to its next position, and a complex head at its final position. Again, triangles are fine for pronouns (including interrogative pro-forms like *where* and *who*) and bagels. An example follows.

Example. How did they eat bagels?



Note on the tree above: This is an example of an adjunct *wh*-word, which is conventionally considered to have the category of an adverb. The *wh*-word corresponds to an adjunct in the answer, either an AdvP (*sloppily*) or a PP (*with a spork*).

If you are using `jssyntaxtree`, you can use the following code for this.

```
[CP [AdvP how .] [C' [C [T [V DO] [T ∅past]] [C ∅qwh] ]
[TP [DP they . ] [T' <T> ->3
[VP
[VP [V' [V eat] [DP bagels .]] ]
<AdvP> ->1]]]]]
```

Your task: Draw trees for each of the following sentences:

(12) What had they eaten?

(13) Who might they introduce to you?

(14) Where should they buy bagels?

(15) Why are they buying bagels?

(16) When will they buy bagels?