# Drawing Deep Structures and Surface Structures

Our theory has gotten more complex, so this is a cheat sheet to remember what we're doing and why. We now have *two* levels of representations for all expressions in any language. The first is the Deep Structure (DS), which represents the original position of all of the components of the sentence: the DPs are where they can get $\theta$-roles from the heads that assign them $\theta$-roles, and the heads are where they project phrases. DSs conform to the rules of X′-syntax and the $\theta$-Criterion. DSs are mapped to Surface Structures (SS) by inserting or moving (or deleting) elements. SS represents the ways different languages satisfy various non-semantic syntactic constraints, such as the Stranded Affix Filter (SAF) ("every affix must have a host") and the Extended Projection Principle (EPP) ("Every clause must have a subject in Spec,TP") and the Case Filter ("Every overt argument DP must check an Abstract Case feature").[1] SSs look a lot like their corresponding DSs, but with a few elements moved around and some others inserted (or deleted). Indeed, a single DS (or several closely related, nearly identical DSs) may result in various SSs:

(1) DS: -ed John drink my beer.

a. SS1: John drank my beer. (Raise subject for EPP/Case, lower T for SAF)

b. SS2: John did drink my beer. (Raise subject for EPP/Case, insert *do* for SAF)

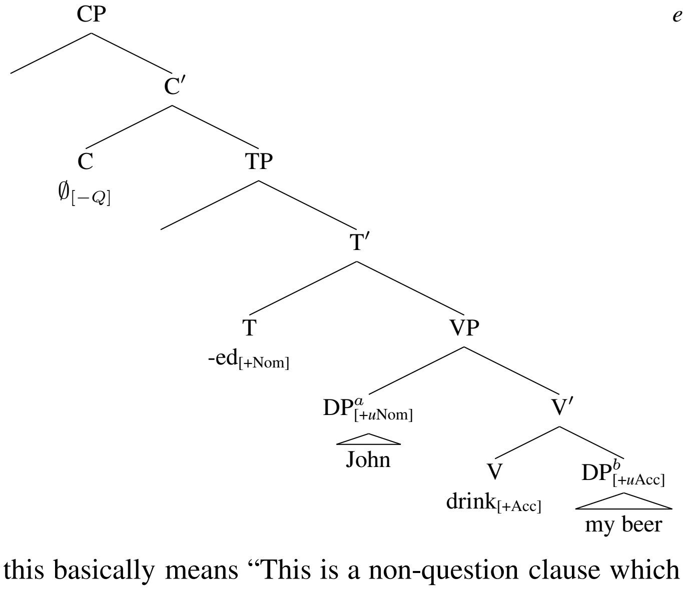
c. SS3: My beer, John drank.
(Raise subject for EPP/Case, lower T for SAF, raise object to emphasize it)

d. ...

All of the SSs in (1) have the same thematic meaning, represented by the DS (John is the agent of drinking the theme in the past). They differ in how that meaning surfaces.

At DS everything is in its original position. The tense affix is in T and the subject is in Spec,VP ("specifier of VP"). Everything conforms to X′-syntax, and the $\theta$-grid of each verb is satisfied in its VP:

(2) DS:



*eat*:

| DP<br>Agent | DP<br>Theme |
|---|---|
| *a* | *b* |

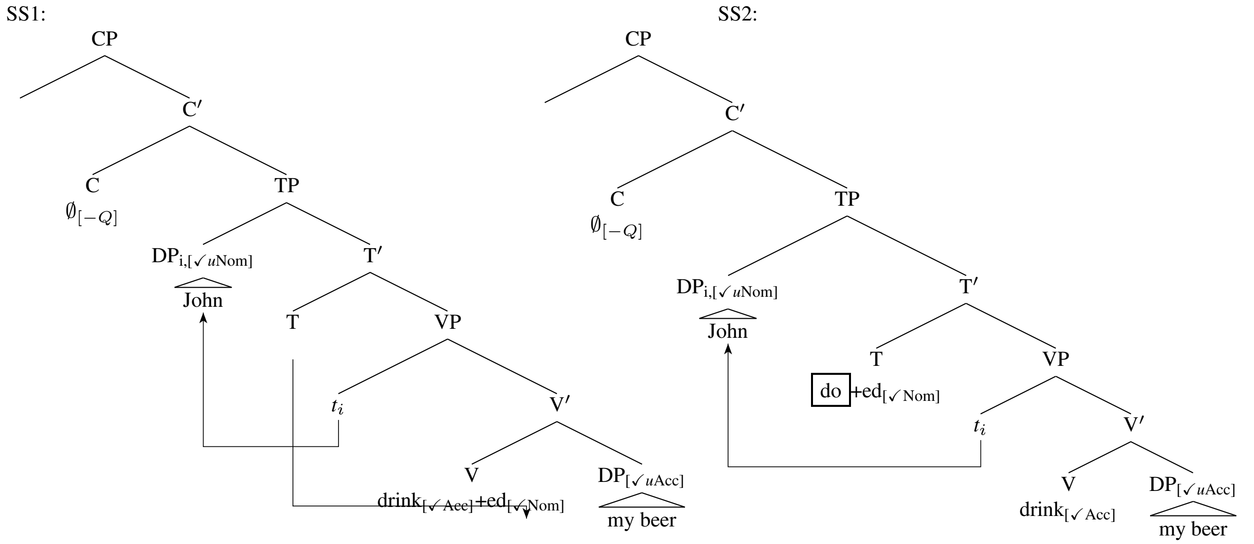
Read top to bottom, this basically means "This is a non-question clause which asserts that in the past John was the agent of drinking and my beer is the theme." Perhaps the two oddest things here are that T is an affix sitting all by itself, and *John* is below it, in Spec,VP. But we have empirical evidence for this:

[1] In fact, there's a deeper principle we're assuming called the Feature Checking Principle that ensures that all uninterpretable features get checked by SS, from which the Case Filter derives as a subtype, and the EPP can be reduced to by assuming Ts have a feature that requires a specifier. I'll maintain the old terms Case Filter and EPP here.

- We know the tense affix is in T position because we know tense morphemes (affixes or words) start out in a different place than Vs (which explains why if you insert negation, move or delete VP, or move the T, the T and V separate, and why they are in complementary distribution with modal auxiliaries).
- We know the subject is base-generated in Spec,VP because this is the position at which *all* gets stranded in *The men all drank my beer*.

When a DS such as (2) is mapped to SS, we must make sure the EPP, SAF, and the Case Filter are satisfied (and more properly that the Feature Checking Principle is satisfied, which we can reduce the Case Filter and the EPP to). These are superficial constraints of English (and lots of other languages). We satisfy the EPP by raising the subject to Spec,TP, which always happens in English, and in this position the subject will also check Case. We satisfy the SAF in English by lowering the affix *or* inserting *do*. Thus we get the following two possible SSs (I mark insertion by drawing a box around the inserted element):

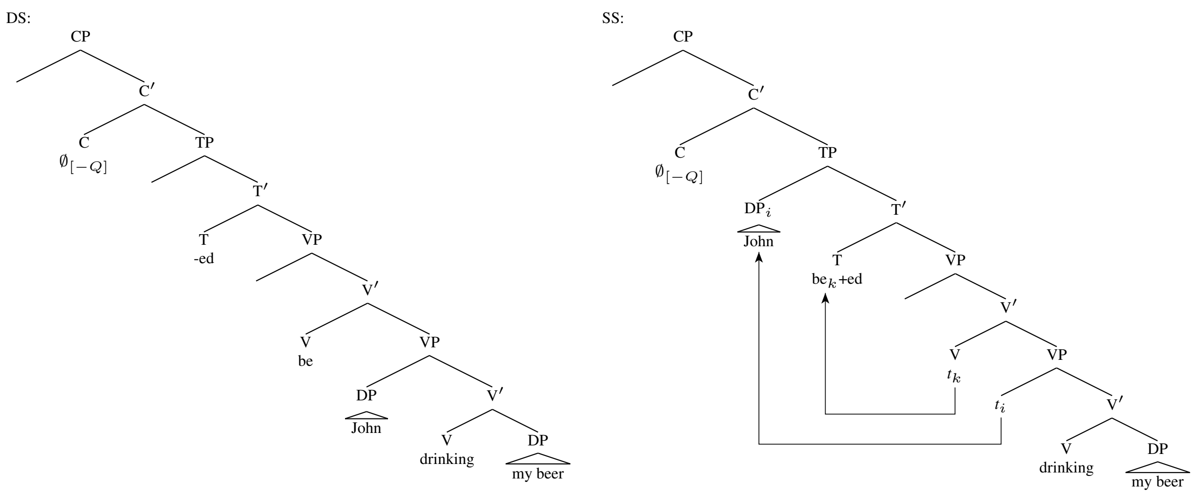
(3)


SS1:
CP
C′
C
$\emptyset_{[-Q]}$
TP
$\text{DP}_{\text{i},[\checkmark u\text{Nom}]}$
John
T′
T
VP
$t_i$
V′
V
~~$\text{drink}_{[\checkmark \text{Acc}]}\text{+ed}_{[\checkmark \text{Nom}]}$~~
$\text{DP}_{[\checkmark u\text{Acc}]}$
my beer
SS2:
CP
C′
C
$\emptyset_{[-Q]}$
TP
$\text{DP}_{\text{i},[\checkmark u\text{Nom}]}$
John
T′
T
$\boxed{\text{do}}\text{+ed}_{[\checkmark \text{Nom}]}$
VP
$t_i$
V′
V
$\text{drink}_{[\checkmark \text{Acc}]}$
$\text{DP}_{[\checkmark u\text{Acc}]}$
my beer

These SSs "Spell-Out" (i.e. are read phonologically) as *John drank my beer* and *John did drink my beer* respectively. This is how we will generate *all* clauses of English, including main and embedded clauses. Note that *all subjects start in Spec,VP at DS and raise to Spec,TP at SS*. This holds for all clauses in English (and most other languages except Irish). If this were French, it would be mostly the same (with different words), but instead of lowering T to V we would raise V to T. There is no French version of *do* (that I am aware of). SS1 in (3) is very common, and a lot of trees you draw will look like it.

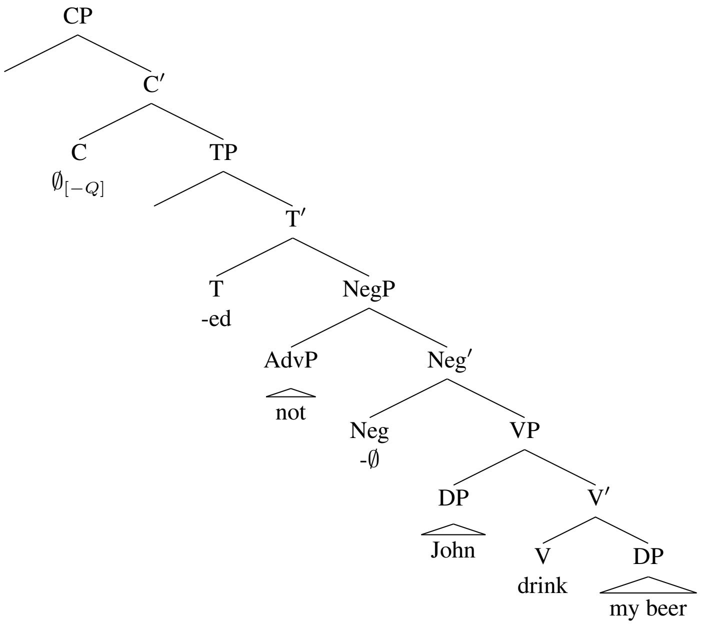
Remember that auxiliary *have* and *be* work a bit different. These start off as Vs and actually do raise to T. They are the only verbs in English that do this, but in French *all* verbs do this (on the next several trees I am leaving off the Case features so they're easier to read):

(4)



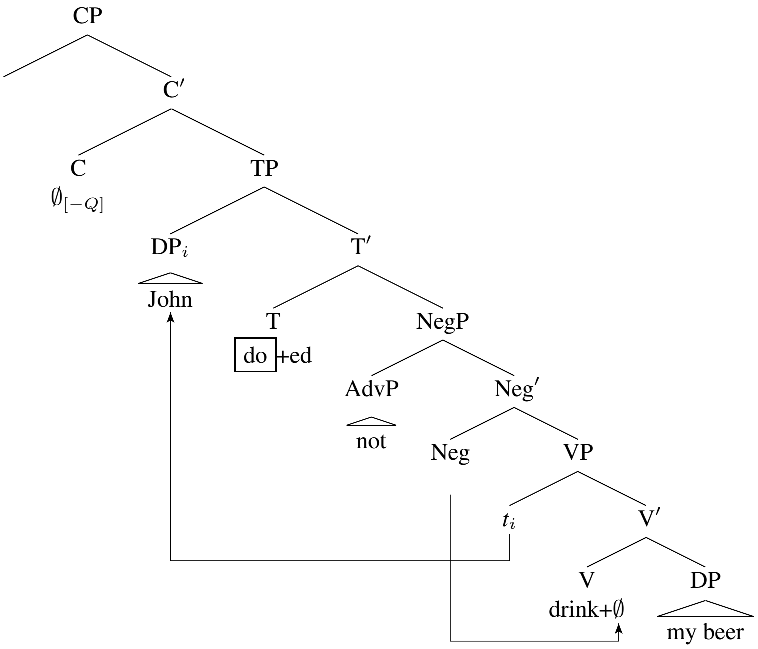
The SS Spells-Out as *John was drinking my beer*. Next, recall that negation forms its own projection between T and V, when it is present. In English *not* is a special Adv that goes in Spec,NegP, while the actual Neg head is a null affix. In French *pas* goes in Spec,NegP, and the Neg head is *ne-*. Here's an English DS with negation:

(5) DS:



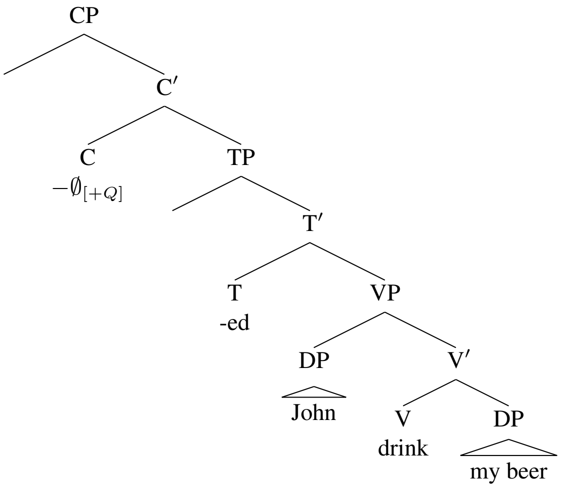
Read top to bottom, this basically means "This is a non-question clause that asserts that in the past it was not the case that John was the agent of drinking and my beer was the theme." We know negation comes here because clausal negation in English (and many other languages) always comes in a special place right after T. We assume that there is a null affix because many languages have affixes (like French, and also Japanese and Korean). In English, a T affix cannot lower to Neg (for mysterious reasons, but perhaps because it cannot lower around a specifier like *not*), so we must insert *do* at SS to satisfy the SAF. The Neg head lowers like other affixes in English.

(6) SS:



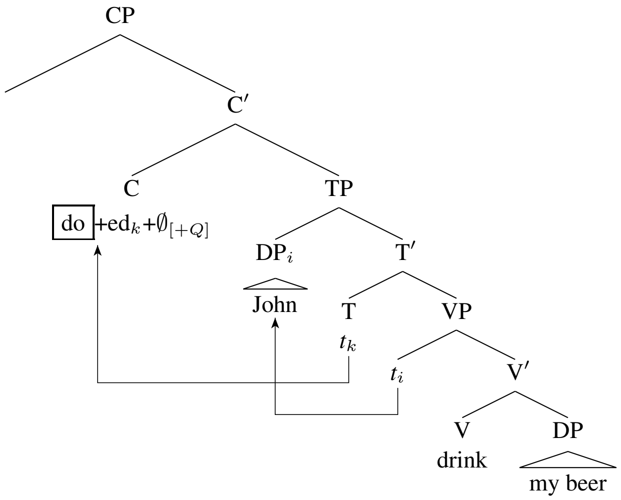
This Spells-Out as *John did not drink my beer*. Finally, we also have yes/no questions, which involve moving T to C, to satisfy the SAF for the $-\emptyset_{[+Q]}$ question affix C (which in Indonesian is realized as *-kah*). At DS a question looks like a declarative sentence, except for the difference in the C (cp. (7) to (2)):

(7) DS:



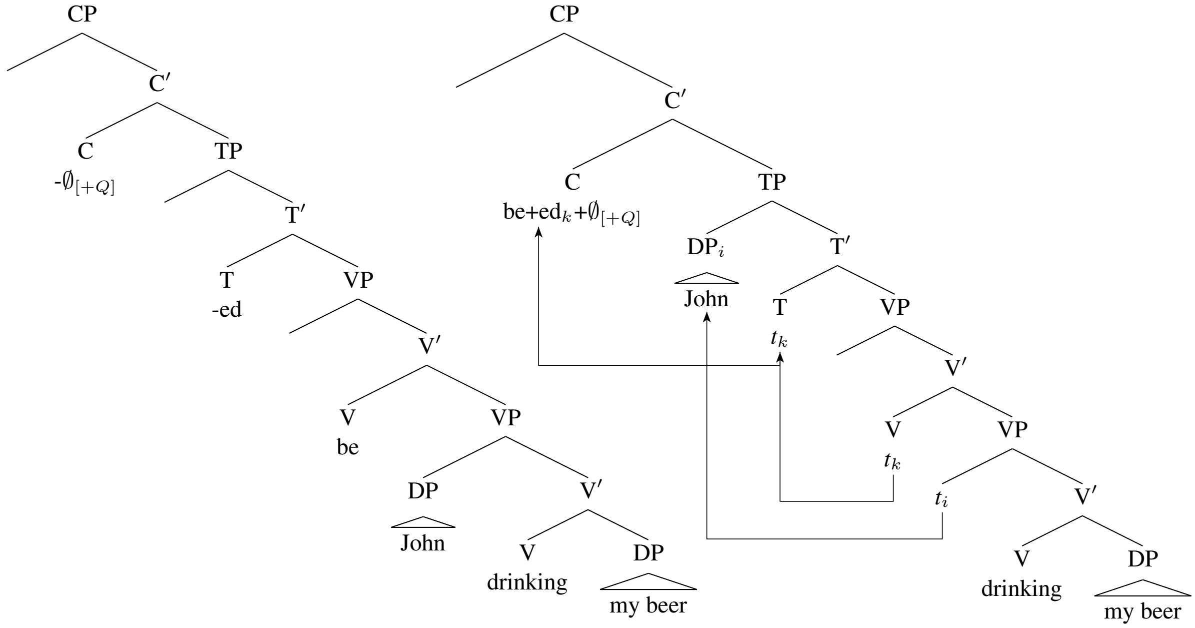
Read top to bottom, this basically means "This is a question clause in which the speaker asks if in the past John was the agent of drinking and my beer was the theme." In this case, the T has two requirements: it must lower to save itself from the SAF, and it must raise to save the C from the SAF. It cannot do both simultaneously, so instead *do* is inserted to save *-ed* from the SAF, and then *do*+*ed* raises to save the C from the SAF, giving us the following SS:

(8) SS:



This Spells-Out as *Did John drink my beer?* However, remember that *have* and *be* are special in English in being able to raise to T. If they raise to T, then they can carry right on and raise to C in a question, meaning we do not have to insert *do*. The following DS is identical to the one in (4), except it is a question (and thus has the special question C). At SS *be* has raised first to T and then to C, saving both affixes from the SAF (and of course the subject raises as it always does in English):

(9) DS: SS:



This SS Spells-Out as *Was John drinking my beer?* These are just *some* of the issues you'll deal with on the homeworks. Note that the order of heads in the clause is evidenced primarily by cases where we do not have affixes. Since on our theory affixes trigger movement, if we look at sentences lacking them (to the degree possible) we can probe the basic word order. The embedded clause in (10) gives us evidence for a basic C-T-Neg-V head ordering in English, with dependents of various sorts draped around each head:

| (10) | John | said | [ | **that** | Bill | **will** | not | $-\emptyset$ | often | **call** | his | mother | ]. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | C | | T | | Neg | | V | | | |

It is usually assumed that the order of heads is the same across languages, but that may not be true.

Above I was showing various DP-movements within a clause, but remember that we can also have movement across clauses if a DP needs Case — infinitive Ts do no check Case on their specifiers, just as Vs do not check Case on their specifiers. So for an infinitival clause whatever DP ends up satisfying the EPP there will have to move further up the tree to check Case, deriving movement out of the clause. The class of verbs that triggers this is raising verbs/predicates like *seem*, *appear*, and *is likely*:

(11) *seem*:

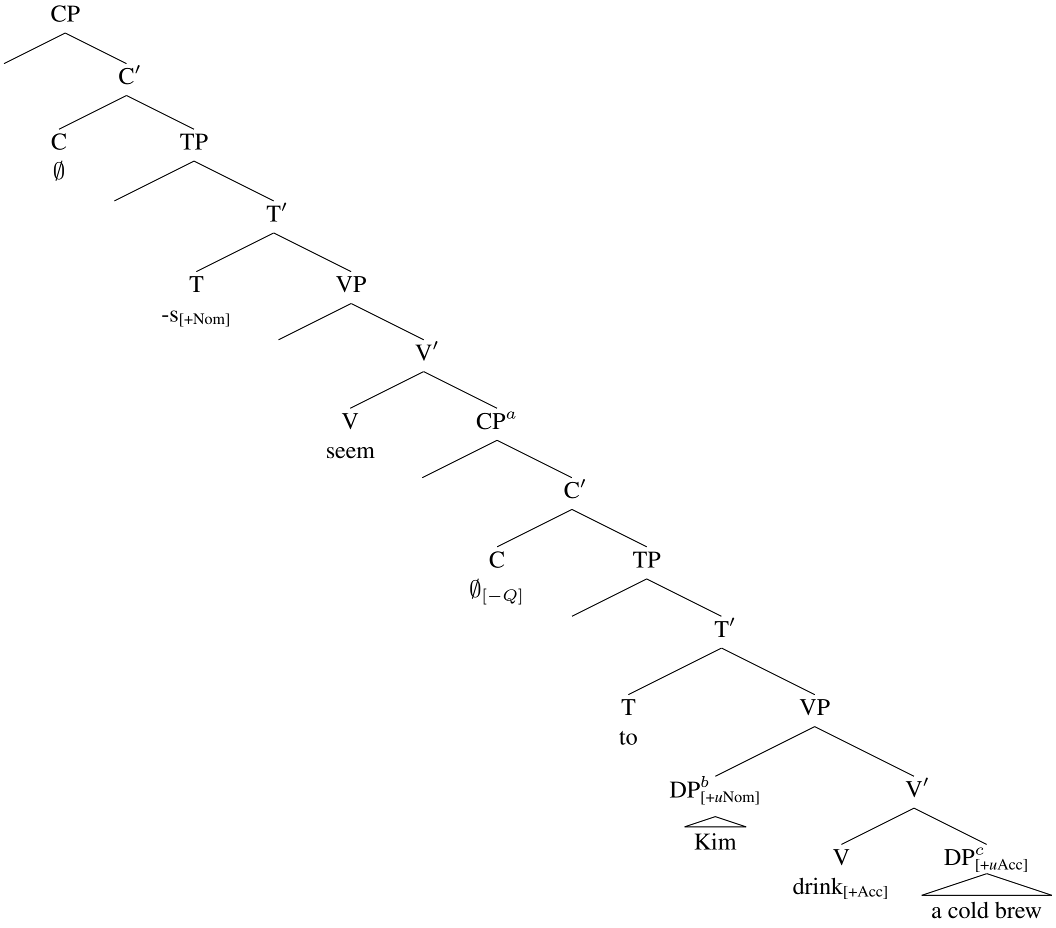
| CP<br>Proposition |
|---|
| a |

*drink:*

| DP<br>Agent | (DP)<br>patient |
|---|---|
| b | c |

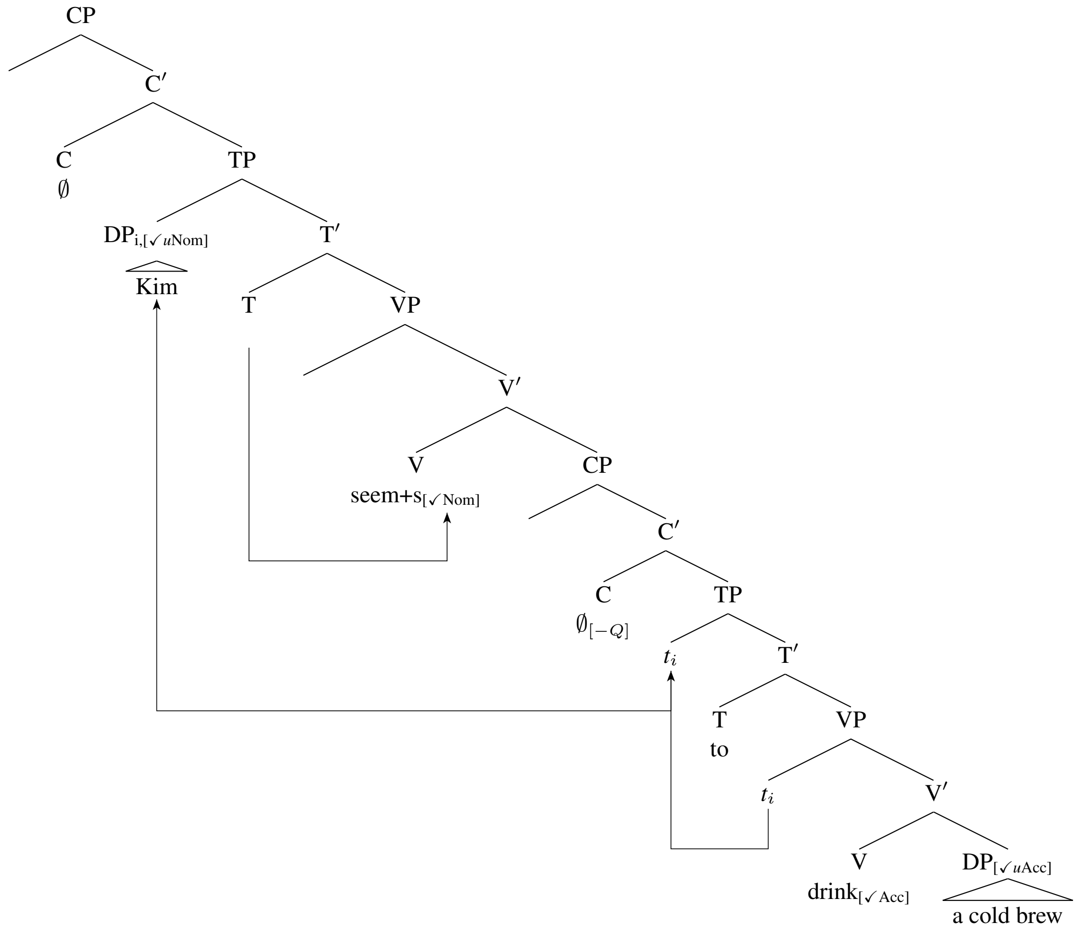
One possible DS for *seem* with a complement clause based on *drink* is this, with an infinitive lower clause:

(12) DS1:


CP
C′
C
∅
TP
T′
T
$\text{-s}_{[+\text{Nom}]}$
VP
V′
V
seem
$\text{CP}^{a}$
C′
C
$\emptyset_{[-Q]}$
TP
T′
T
to
VP
$\text{DP}^{b}_{[+u\text{Nom}]}$
Kim
V′
V
$\text{drink}_{[+\text{Acc}]}$
$\text{DP}^{c}_{[+u\text{Acc}]}$
a cold brew

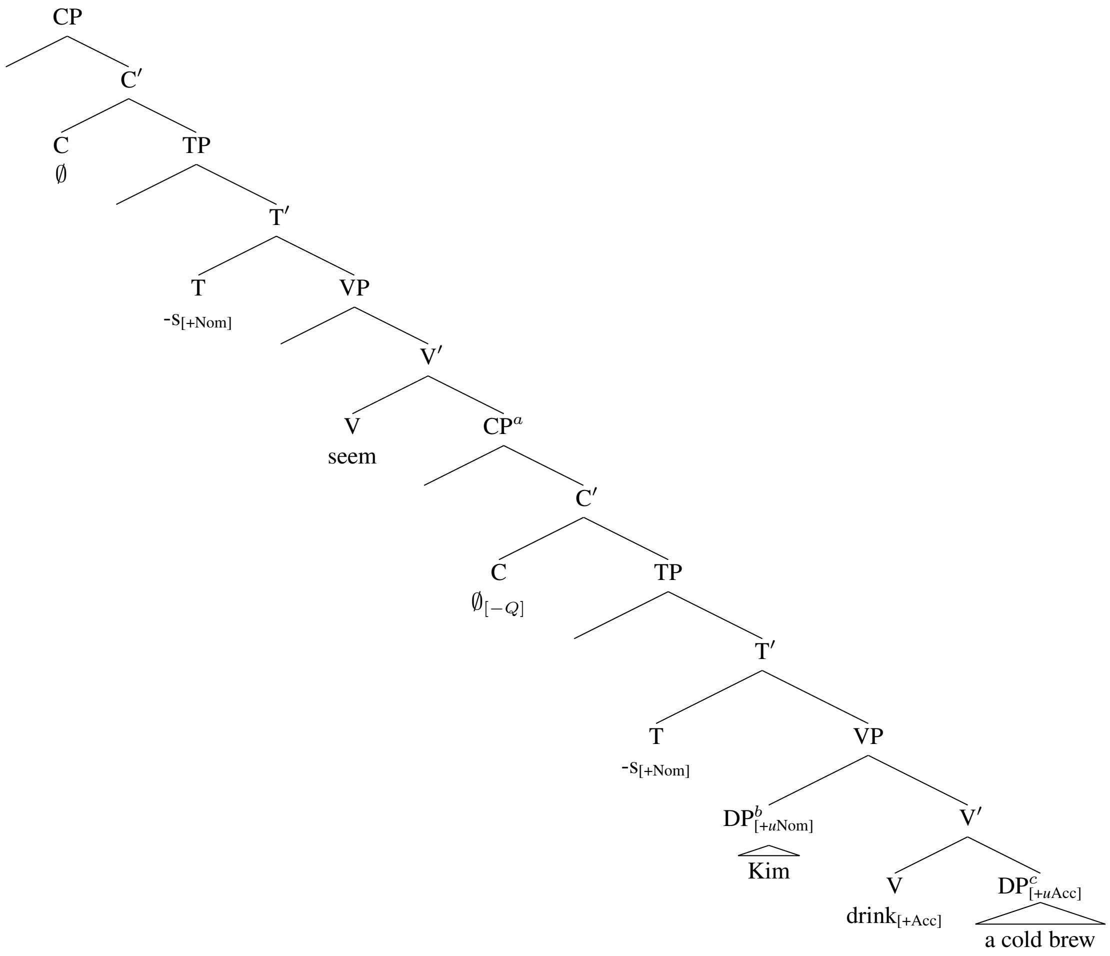
Both clauses need a subject to satisfy the EPP, and the lower subject needs Case, but only the top T has one. So that DP raises first to the lower Spec,TP (for EPP reasons) and then the upper one (for EPP and Case reasons)x:

(13) SS1:


CP
C′
C
∅
TP
DP$_{\text{i},[\checkmark u\text{Nom}]}$
Kim
T′
T
VP
V′
V
seem+s$_{[\checkmark \text{Nom}]}$
CP
C′
C
$\emptyset_{[-Q]}$
TP
$t_i$
T′
T
to
VP
$t_i$
V′
V
drink$_{[\checkmark \text{Acc}]}$
DP$_{[\checkmark u\text{Acc}]}$
a cold brew

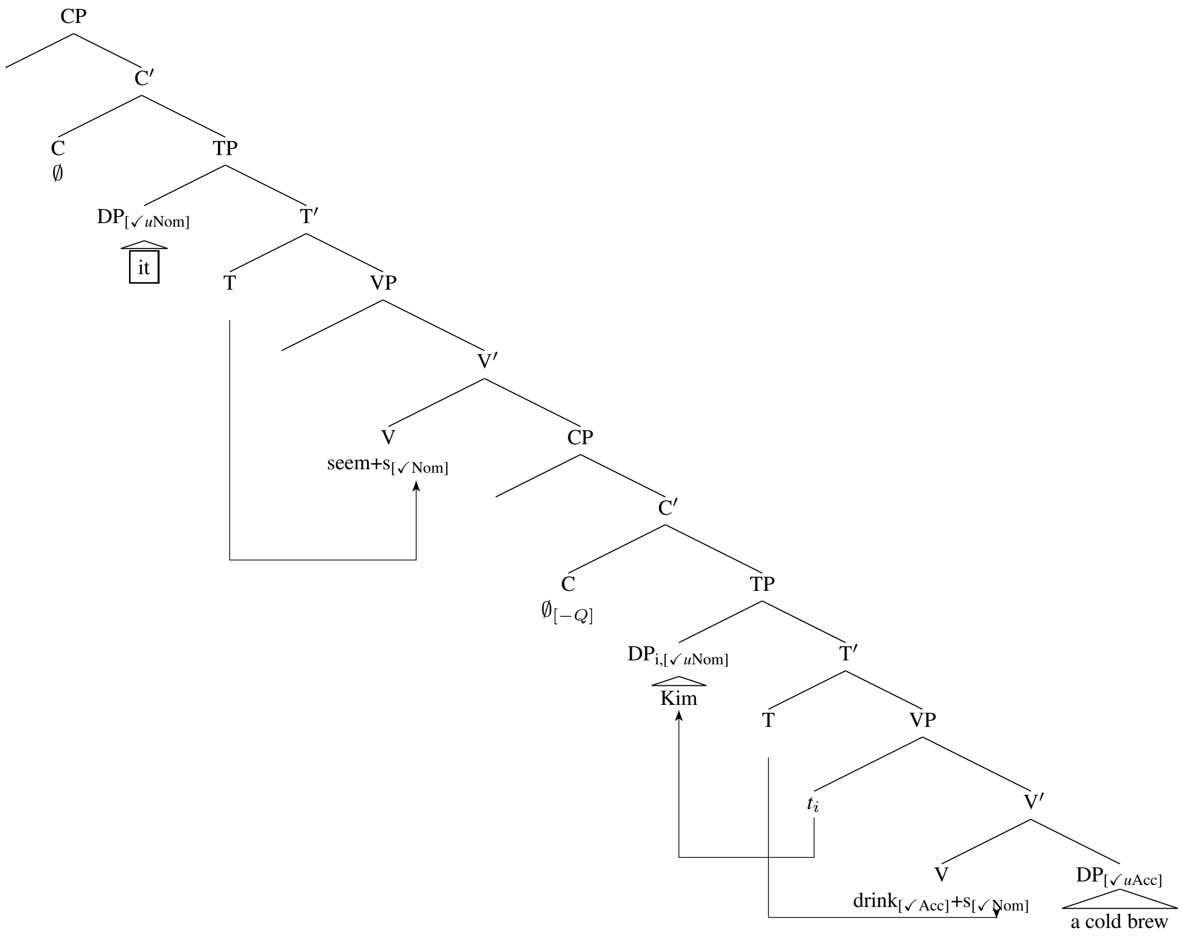
The other DS is this, with the lower clause being finite rather than infinitive:

(14) DS2:



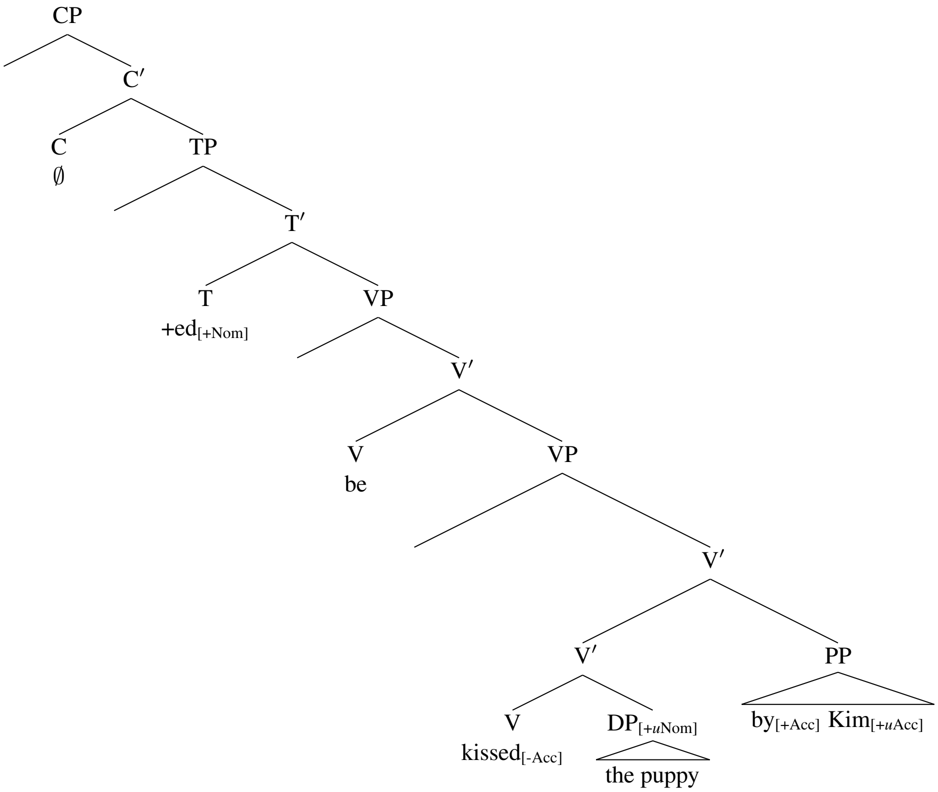
Here the lower subject can stop moving once it hits the the lower Spec,TP, since it can get case there since T is finite. In this case we insert an expletive upstairs (which also checks Case) as a way to satisfy the EPP:

(15) SS2:

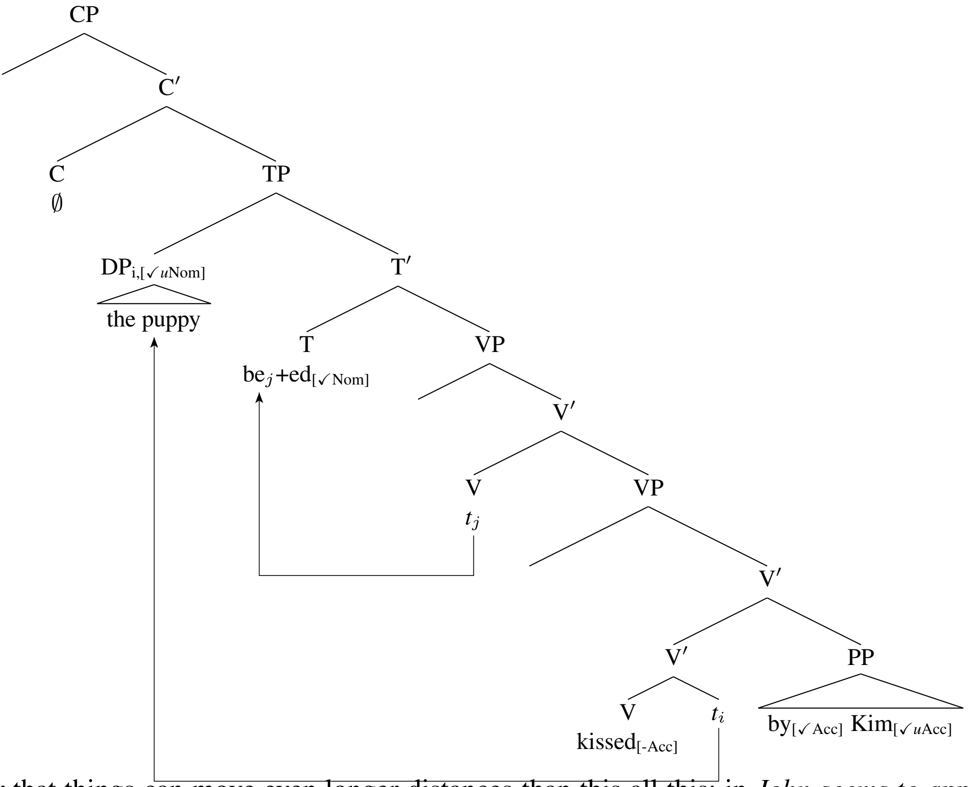


Above we were looking at subject DPs of various sorts moving, whereas object DPs usually don't move because they can check Case right off of the verb. But remember that sometimes things that start off as complements can't check Case, as with passives (where passivization stops a V from checking Accusative). In this case the object DP must raise:

(16) DS:



(17) SS:


CP
C′
C
∅
TP
DP$_{\text{i},[\checkmark u\text{Nom}]}$
the puppy
T′
T
be$_j$+ed$_{[\checkmark\text{Nom}]}$
VP
V′
V
$t_j$
VP
V′
V′
V
kissed$_{[\text{-Acc}]}$
$t_i$
PP
by$_{[\checkmark\text{Acc}]}$ Kim$_{[\checkmark u\text{Acc}]}$

Finally, remember that things can move even longer distances than this all this: in *John seems to appear to have been kissed* the subject *John* starts as the object of *kissed*, raises first to the lowest Spec,TP, then next lowest Spec,TP, and finally to the top Spec,TP, all in search of Case (while satisfying the EPP along the way).

On another note, remember that in control structures such as *I want to leave* there is a PRO in the lower clause. Remember that PRO is like any other DP: it can also raise if need be, though not to check Case. But it will raise to satisfy the EPP. This means if it starts in Spec,VP of an active verb or Comp,V of a passive verb it will raise to Spec,TP to satisfy the EPP (I use bracketed structures here for simplicity, but please remember these are just shorthands for trees):

(18) PRO raising from Spec,VP to Spec,TP to satisfy EPP:

a. DS: [$_{\text{CP}}$ -s he$_i$ want [$_{\text{CP}}$ to PRO$_i$ leave ] ]

b. SS: [$_{\text{CP}}$ he$_i$ $t_i$ want-s [$_{\text{CP}}$ PRO$_i$ to $t_i$ leave ] ]

(19) PRO raising from Comp,V to Spec,TP to satisfy EPP:

a. DS: [$_{\text{CP}}$ -s he$_i$ $t_i$ want [$_{\text{CP}}$ to be seen PRO$_i$ ] ]

b. SS: [$_{\text{CP}}$ he$_i$ $t_i$ want-s [$_{\text{CP}}$ PRO$_i$ to be seen $t_i$ ] ]

Crucially, PRO can even raise out of its own clause to satisfy the EPP:

(20) PRO raising from lower Spec,TP to higher Spec,TP to satisfy EPP:

a. DS: [$_{\text{CP}}$ -s he$_i$ want [$_{\text{CP}}$ to appear [$_{\text{CP}}$ to be seen PRO$_i$ ] ] ]

b. SS: [$_{\text{CP}}$ he$_i$ $t_i$ want-s [$_{\text{CP}}$ PRO$_i$ to appear [$_{\text{CP}}$ $t_i$ to be seen $t_i$ ] ] ]

Here, PRO started off in Comp,V of the lowest verb (getting a theme $\theta$-role), and then it raised first to the lowest Spec,TP to satisfy the EPP, and then the next Spec,TP to satisfy the EPP for that clause, and then stopped, since there was no more EPP for it to satisfy (this is satisfied in the top clause by *he*, which moved to the highest Spec,TP to check Case).[2]

Here are some summaries, hints, and additional details:

- You can think of the main VP at DS as a little thematic core of the sentence: no tense, no aspect, just a pure concept of who did what to whom. The Neg, T, and C just add additional information about temporality, speech act (assertion vs. question), etc. on top of that thematic core.
- You can think of the SS of a given sentence as what happens when that nice thematic core faces the realities of the syntactic quirks of particular languages, and things shift around a bit to conform to a few additional constraints.
- At DS, the subject of *each* clause is always in Spec,VP. At SS, the subject of *each* clause always raises to Spec,TP in English and French, but not Irish.
- T affixes always start out under T in DS and must attach to the V through some movement or insertion at SS. English and French differ in how this is done:
  - English T lowers to V, although *have* and *be* raise to T. All French Vs can raise to T.
  - In English you can also insert *do* to satisfy the SAF for T.
- We have evidence for the English vs. French contrast from empirical data: adverbs that come between T and V at DS come before the V at English SS and after the V at French SS.

[2] You may wonder why PRO needs to raise at all. Since it doesn't need to check Case, we could just satisfy the EPP by expletive insertion, generating **He$_i$ wants it to be seen PRO$_i$* meaning "He wants to be seen". There are two problems with this. First, *it* also needs Case, and isn't getting it here. Second, it is usually assumed that PRO in these constructions is like an anahpor, and needs to be in Spec,TP to be bound by an antecedent in the next highest clause (where the binding domain for PRO is a bit bigger than regular anaphors, but is like traces). None of these details are relevant for us here.

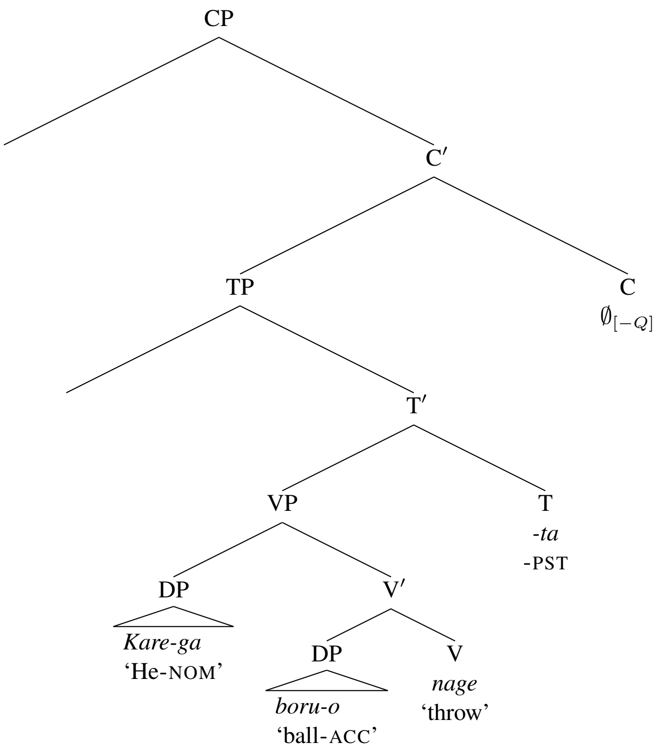
- A question is marked by a special question C at the top of the clause. In English, French, and Indonesian, this can be an affix that must be saved from the SAF:
  - In French, V raises first to T and then to C.
  - In English, *do* is inserted into T and then T raises to C.
  - Alternatively, English *have* and *be* can raise from V to T to C.
- In Indonesian the question C can also be a word *apa*. This is not an affix, so no movement is required.
- Overt argument (specifier and complement) DPs must check Abstract Case, and will move if they can't check them in their DS position.
- PRO is there when another DP must take a theta-role that isn't present on the surface. It is like any other DP in every respect — it can bind things, fill positions, take theta-roles, move — save that it doesn't need to check Case (or it checks some weird other Case nothing else checks). Either way, it can sit in Spec,TP for an infinitival T.

Most importantly, the structures above (for English) are a lot of the clause types we will see in the class. As you draw DSs and SSs, confer with the trees on this handout to make sure things are in the right places.

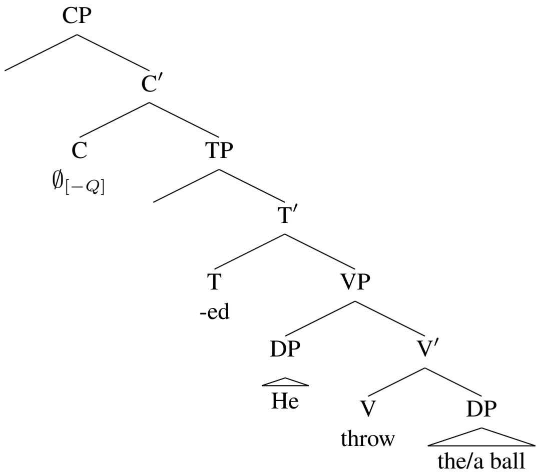
Finally, one other thing to mention: in languages that are not fundamentally SVO, the word order is explained by ordering parameters in X′-theory. This means, for example, that Japanese, which is an SOV language, has structures in that differ from English at DS. For example, the DS for the sentence in (21a) is as in (21b), compared to the DS for the English translation in (21c):

(21) a. *Kare-ga boru-o nage-ta*
He-NOM ball-ACC throw-PST
‘He threw the/a ball

b.



c.



The difference is simply that the order of heads and complements for each XP is reversed.