Ling 165B: Syntax II

Nico(letta) Loccioni

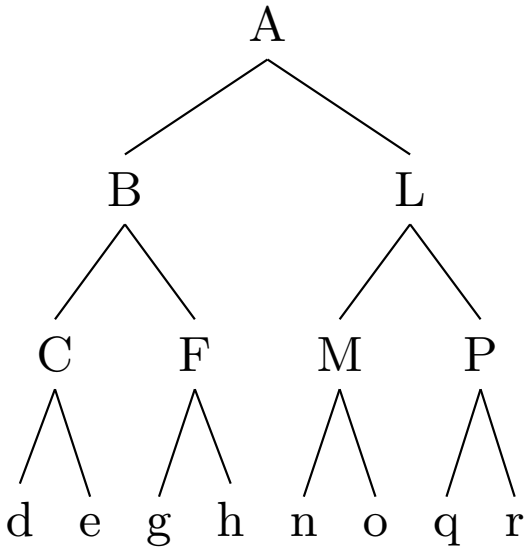
January 05, 2022

Structural relations

C(onstituent)-Command:
Node A c-commands node B if

(i) every node dominating A also dominates B;

(ii) and A does not itself dominate B



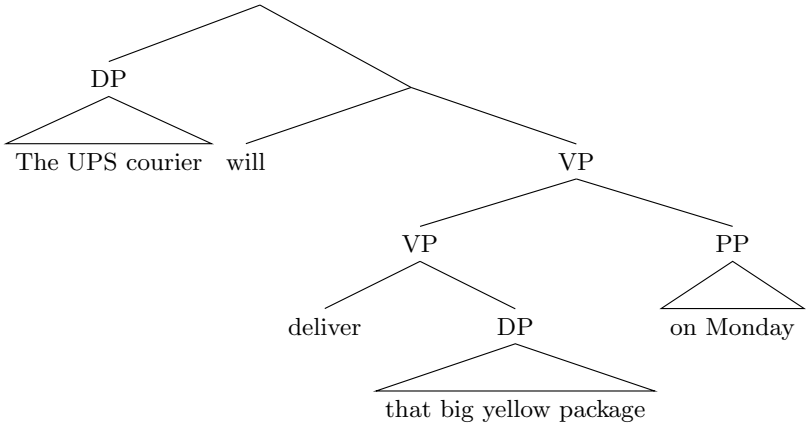
Asymmetric C-Command:
A asymmetrically c-commands B if A c-commands B but B does NOT c-command A.

Exhaustive dominance
A node α exhaustively dominates a set of terminal nodes N iff (i) α dominates every node in N (so that there is no member of the set N that is not dominated by α), and (ii) α does not dominate any terminal node not in N.

- Exhaustive dominance is useful to talk about the leaves of a tree (which are the lexical items).

- Leaves form a *constituent* precisely when there is a node in the tree to exhaustively dominate those leaves.

Take the tree from last time:



(i) {deliver, that, big, yellow, package} forms a constituent because there is a node that exhaustively dominates these leaves. Which one?

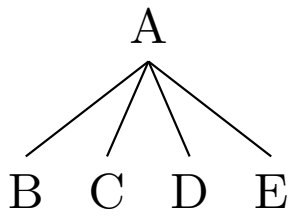
(ii) {yellow, package, on Monday} does not form a constituent because...

NOTA BENE: immediate domination is not a necessary condition for exhaustive dominance.

In tree (1), A exhaustively dominates B, C, D and E because:

- B, C, D and E are terminal nodes
- A dominates all of them
- there is no other terminal node dominated by A (i.e B, C, D and E are the only terminal nodes dominated by A.

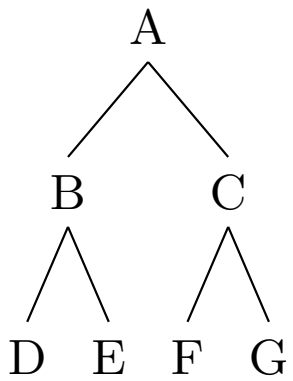
(1)



In structure (2), A exhaustively dominates D, E, F, G because:

- D, E, F and G are terminal nodes
- A dominates all of them (it does not matter that it does not immediately dominate them)
- there is no other terminal node dominated by A (i.e D, E, F and G are the only terminal nodes dominated by A.

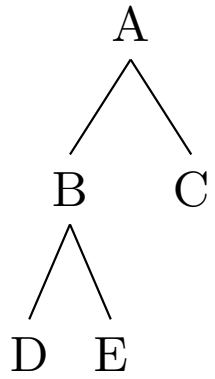
(2)



In structure (3), A exhaustively dominates D, E and C but it does not exhaustively dominate D and E, because:

- it is true that D and E are terminal nodes
- it is also true that A dominates all of them (it does not matter that it does not immediately dominate them)
- but there is another terminal node (namely C) which is dominated by A.

(3)



Practice Draw a tree with the following properties:

- non-binary branching nodes are forbidden;
- A is a root node;
- f, g, h, i, l and m and g are all terminal rodes;
- A, B, C, D and E are not-terminal nodes;
- B sister-precedes E;
- D asymmetrically c-commands B;
- D sister-precedes C;
- A does not immediately dominate B;
- D exhaustively dominates { m, l }
- m precedes f;
- i precedes g;
- l sister-precedes m;
- g is not i's sister;
- E dominates h.
- i does not sister precede f
- g precedes h

Structural ambiguities

A sentence can have more than one meaning. It's called 'ambiguity'

It comes in two flavors:

(i) **Lexical ambiguity** uses words with more than one meaning.

(4) Peter went to the bank.
   a. ≡ Peter went to the financial institution.
   b. ≡ Peter went to the side of the river.

There are two lexical entries for 'bank': $bank_1$ and $bank_2$.

(ii) **Structural ambiguity** ambiguous due to the structure or constituency of the sentence.

The relationship between structure and meaning is systematic!

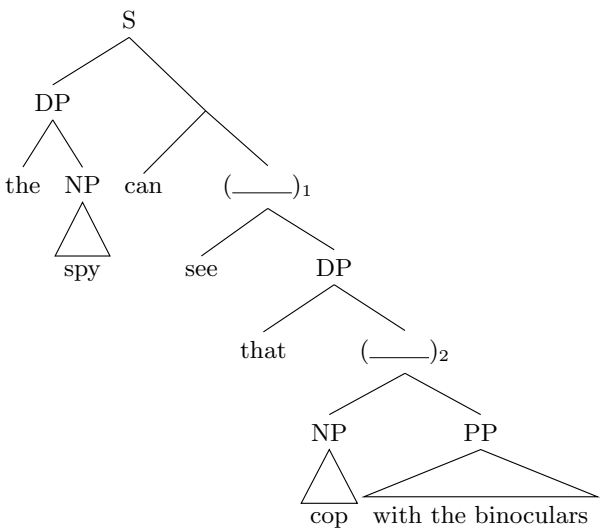
Example

(5) The spy can see that cop with the binoculars.

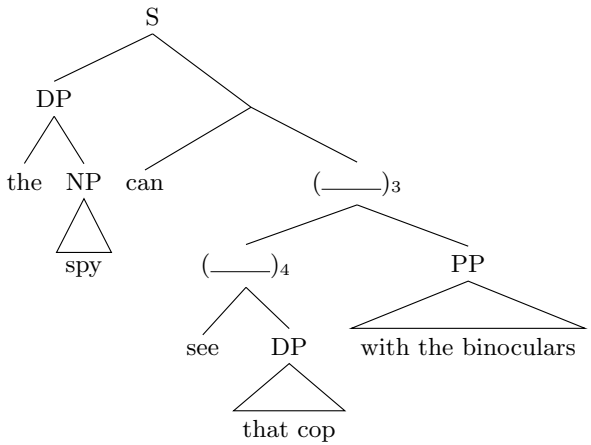
a. ≡ _____

b. ≡ _____

Tree 1



Tree 2



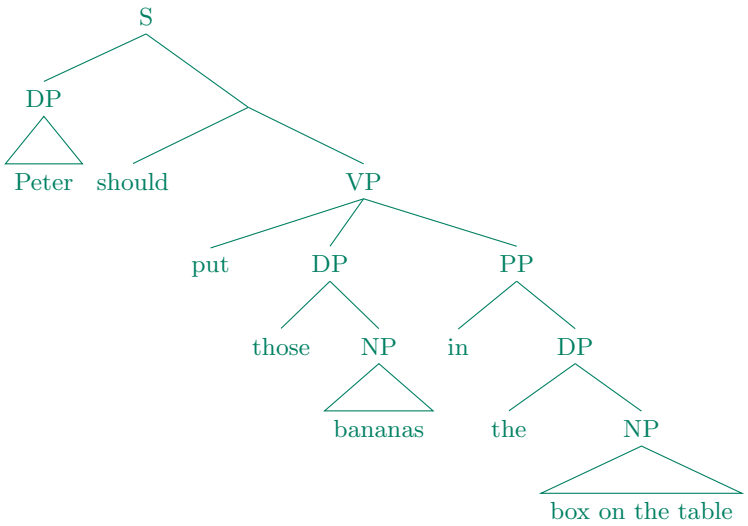
NB: Each tree has an unambiguous interpretation!

- Pair each tree up with the meaning of each sentence;
- Label the nodes 1,2,3 and 4;
- Explain how at least one of the tests supports the structure in the first tree but not the second and how at least one test supports the structure in the second tree but not in the first.

For you to practice!

Consider the following ambiguous sentence:

(6) Peter should put those bananas in the box on the table.



- Draw the other tree structure.
- Pair each tree up with the meaning of each sentence;
- Explain how at least one of the tests supports the structure of one tree but not the other.

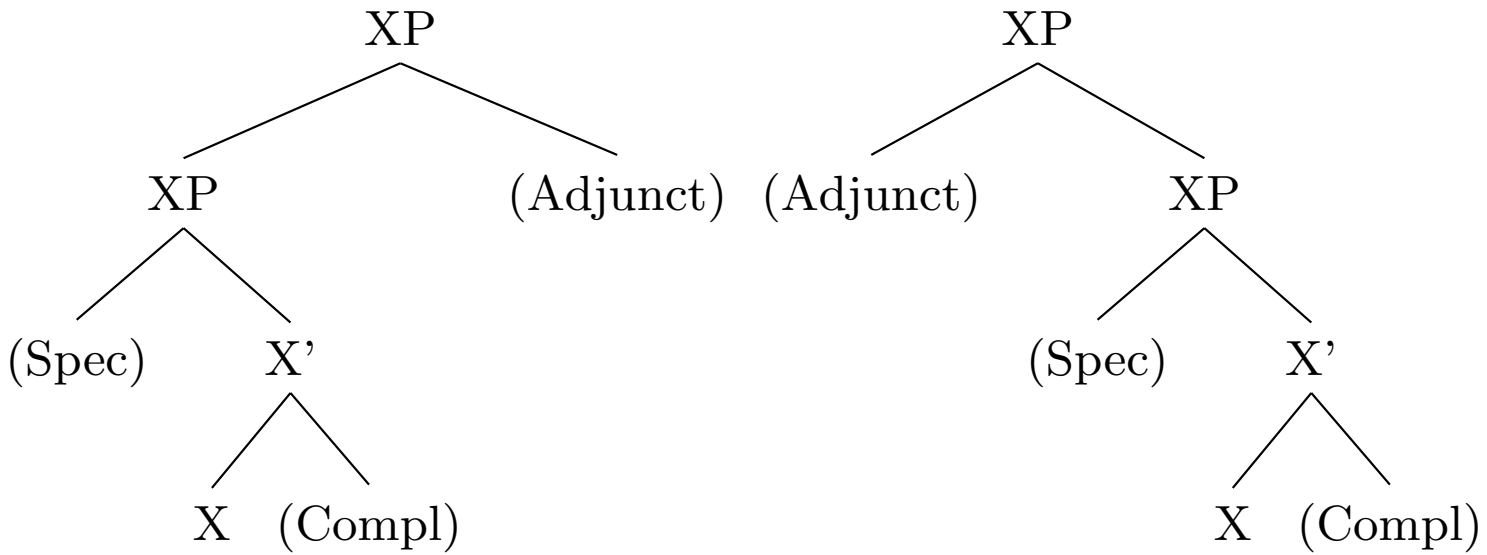
Coming up: X-bar theory

- We know that sentences exist! (So far, we have been using the label 'S' for sentences)
- Based on constituency tests, we have been able to identify:
  - Verb Phrases (VPs);
    do so replacement test; topicalization; pseudo-clefting; coordination; VPE, ...
  - Determiner Phrases (DPs);
    replacement using a pronoun; topicalization; clefting; pseudo-clefting; coordination ...
  - Noun Phrases (NPs);
    substitution by 'one(s)'; coordination; ...
  - Prepositional Phrases (PPs);
    replacement test (there, then...); topicalization; clefting; coordination: NPE...

  We also came across other parts of speech (adjectives, adverbs), modal verbs (should, will, must, can...)

Behind the range of diverse constructions that English and other languages allow, we find surprising uniformity and regularity.

- We are now going to introduce the idea that the rule system that underlies our phrases is very simple.
- Every phrase looks the same!
- For English, syntax will be the iteration of:



This is called **X-bar Theory**.

X-bar Theory

X-bar theory

X-Bar theory is a theory that tries to eliminate phrase structure idiosyncrasies: every phrase looks the same.

- Every phrase is the projection of some lexical category: N, V, A, Adv...
- The system of projected constituency is common to all categories.

We need to distinguish between 4 types of elements:

- **heads**
- **complements**
- **specifiers**
- **adjuncts**

Head

→ Every phrase contains exactly one head of some category (N, V, Adj, Adv, D...)

→ Every head is contained within a phrase of the same category.

→ The head is present in all cases. Whether a complement or a specifier is needed is a case-by-case thing.

→ The head of a constituent projects its label (lexical category) to that constituent.

→ The head of a constituent tells us the distribution of that constituent.

→ The head selects its sister constituent.

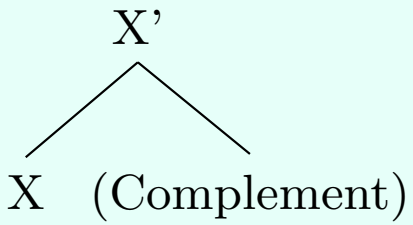
Complements I

→ Complements are selected by the head as their sisters.

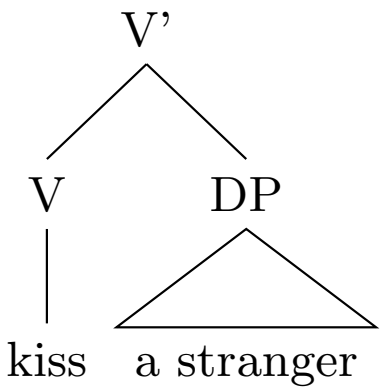
→ They are merged into the structure as a result of the complement rule.

Complement Rule (for English and head-initial languages)

X' → X (Complement)



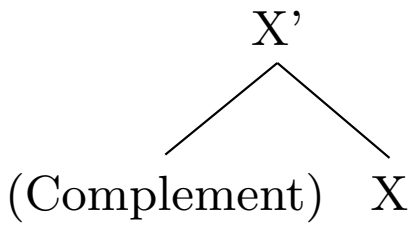
Example



Complements II

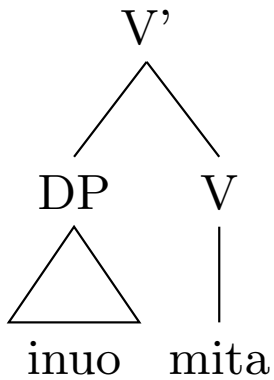
Complement Rule (for head-final languages)

X' → (Complement) X



Example

(7) taro-wa inu-o mita
taro-TOP dog-acc saw
'taro saw the/a dog' (**Japanese**)

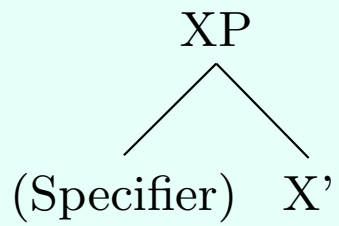


Specifiers

→ Specifiers are merged into the structure as the sister of X'.

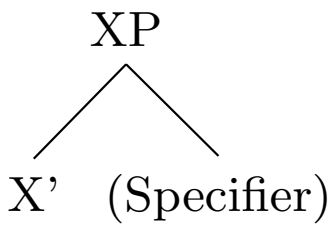
Specifier Rule (for English)

XP → (Specifier) X'



Specifier Rule (for languages such as Malagasy)

XP →X' (Specifier)



Adjuncts

→ Adjuncts are modifiers which modify the phrase.

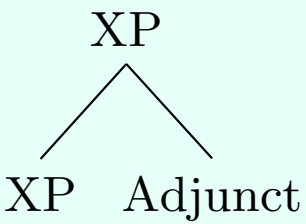
→ They are sisters to phrases (XPs, not X' or X!). If you used Carnie's textbook, you'll need to adjust to this different adjunction level!

→ They tend to be optional.

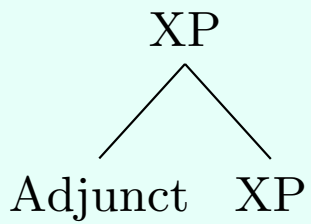
→ In English they can both precede and follow their sister.

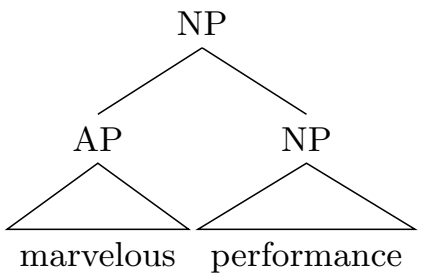
Adjunct Rule.

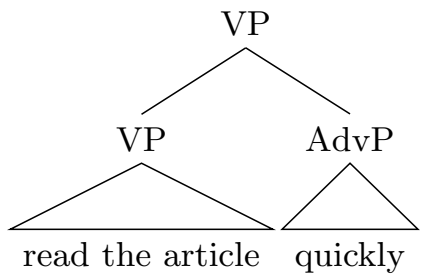
XP → Adjunct XP



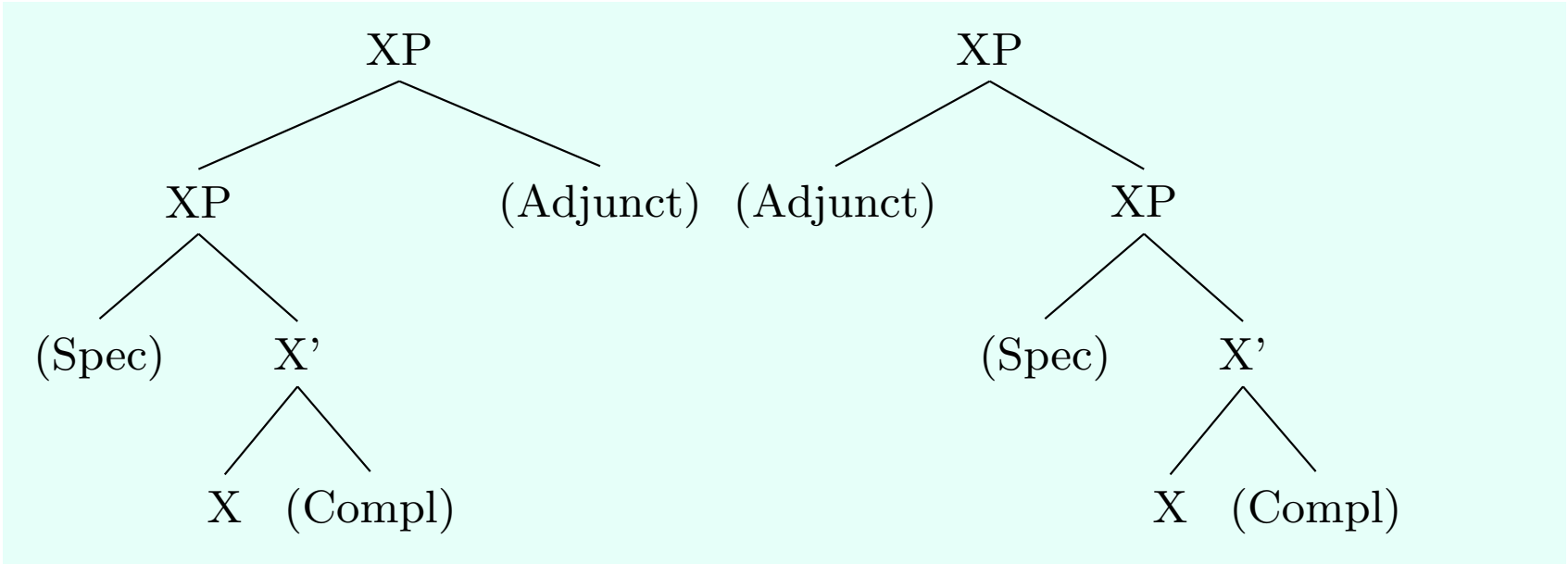
XP → XP Adjunct







This means that the English syntax is the iteration of:



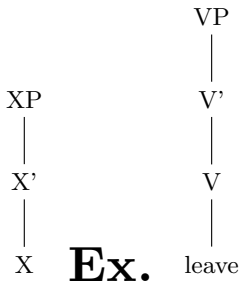
How do we define complements, specifiers and adjuncts? We define them based on their structural position!

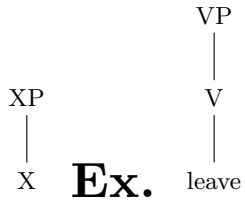
(i) **complement**: sister to X.

(ii) **specifier**: sister to X' , daughter of XP.

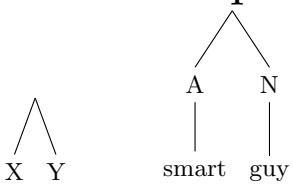
(iii) **adjunct**: sister to XP , daughter of XP.

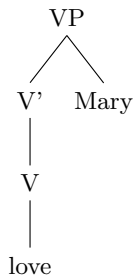
How to write a X-bar compliant tree:

- For every head you want to have three layers: 

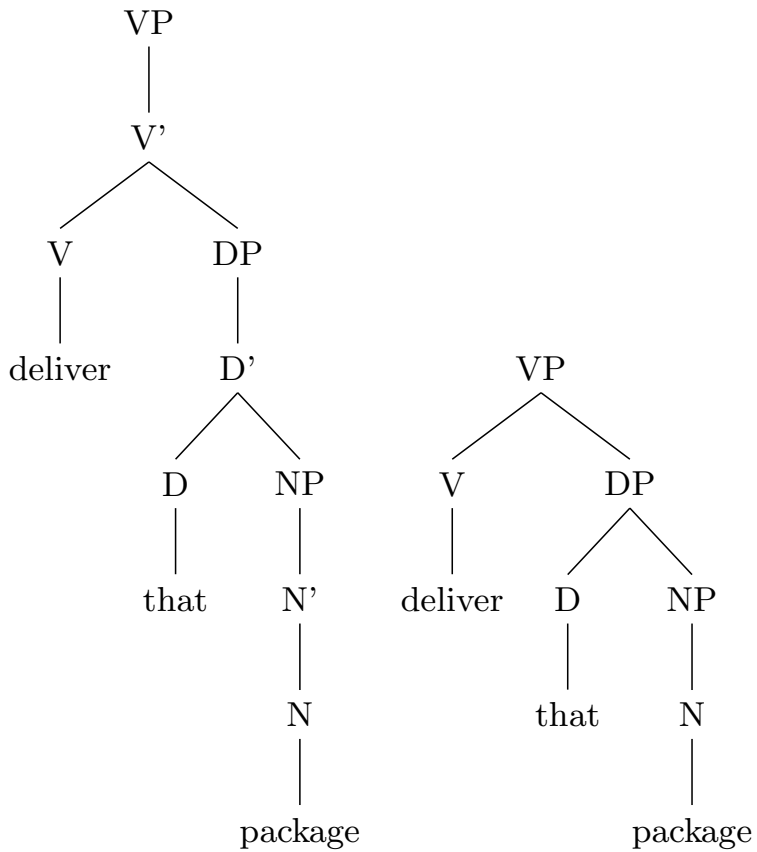
- If there is no specifier, you can omit the X-bar level: 

- Both the specifier and the complement are phrases. Don't draw trees like this:



- Complements are sisters of the head that selects them. Don't attach complements to higher projections of the head! 

So [$_{VP}$ deliver that package] now looks like one of these:



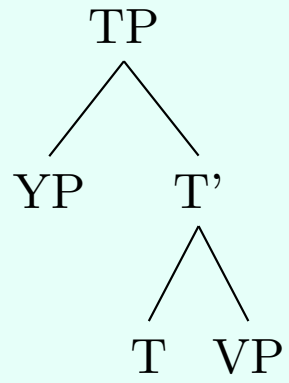
Poll ?? Do you understand why these two structures are equivalent?

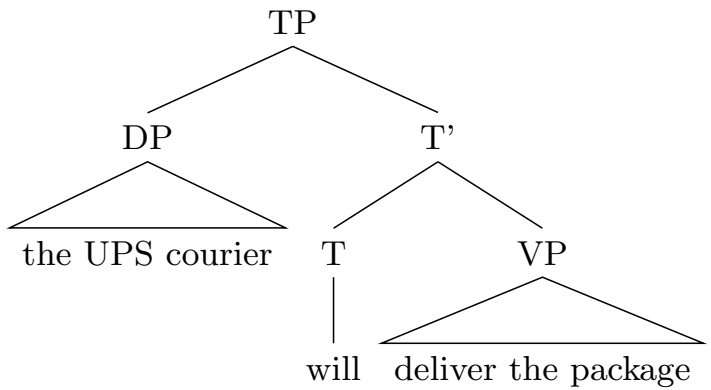
Tense Phrases (TPs) I

- We've been using S for sentences. That does not seem to follow the X-bar schema.
- We are going to adopt a new system where Tense is the head of sentences.

TP:

- The head: T
- Complement: VP
- Specifier: The subject (DP or CP)





Tense Phrases (TPs) II

Realization of T

- T can be **free**
  - 'to' in non-finite clauses (raising to subject, subject control...)

    (8) Mary wants to leave.
  - modals in finite sentences

    (9) Mary should leave.
- T can be **bound**
  - [+pres] (habitual readings, on going interpretations...)

    (10) Mary runs every morning.
  - [-pres]

    (11) Mary danced.

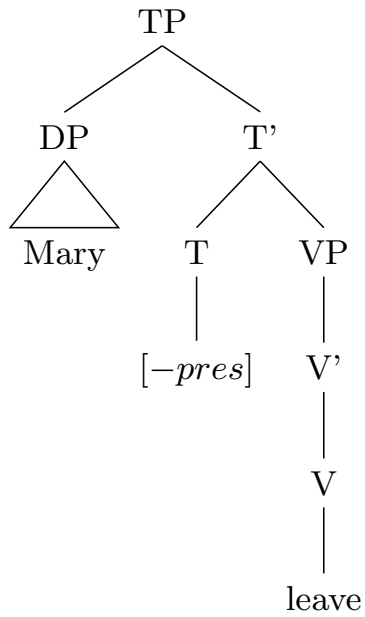
Tense Phrases (TPs) III

Bound morphemes in tree structures

- the bound morpheme is in T;

+pres or [-pres] (-*ed*, -*s* are also fine!)

- the VP is headed by the bare verbal form.



Tense Phrases (TPs) IV

Here are the lexical entries:

to	T	free	selects DP/CP	c-selects VP		
will	T	free	selects DP/CP	c-selects VP	meaning:	future
[+**pres**]	T	bound	selects DP/CP	c-selects VP	meaning:	present
[**-pres**]	T	bound	selects DP/CP	c-selects VP	meaning:	past

Lexical entries are part of the speaker's lexicon. They specify the basic properties of the syntactic atoms (heads), including the category and all the selectional properties. Only unpredictable properties that the speaker has to acquire should be included!

Selection = the type of phrase that the head requires in its specifier (or subject) position.

C-selection = complement selection. It tells us what kind of sister a particular head is selecting for.

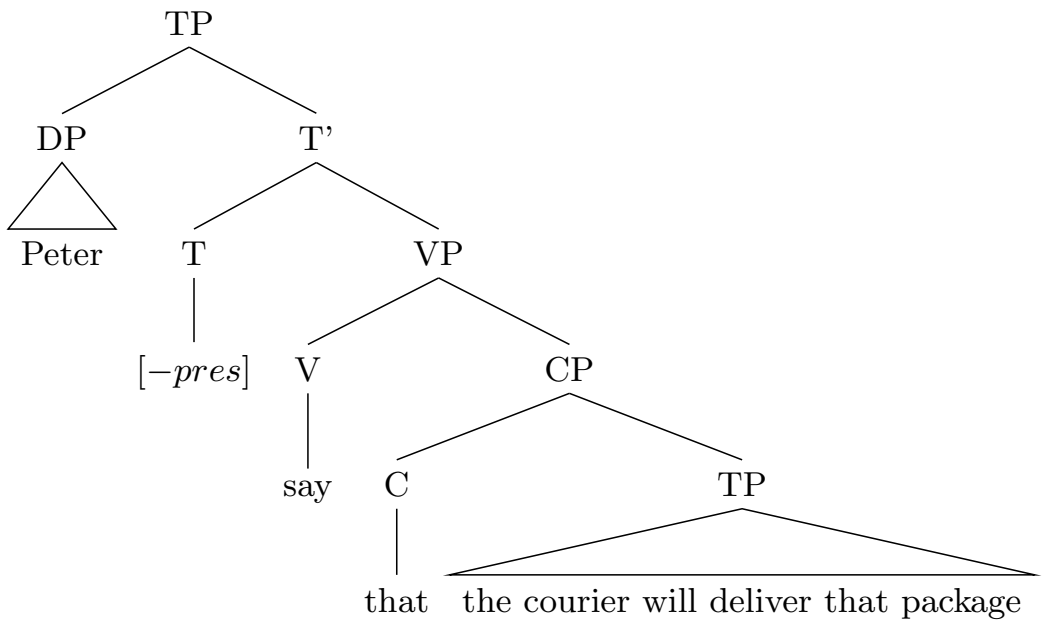
Complementizer Phrases (CPs) I

- Sentences can be embedded under others. Recursion is one of the core properties of human language.

  (12) Peter said that [$_{\text{TP}}$ the UPS courier will deliver that package]

- We saw that we can use *so* replacement in these cases:

  (13) Peter said <u>that the UPS courier will deliver that package</u> → Peter said **so**

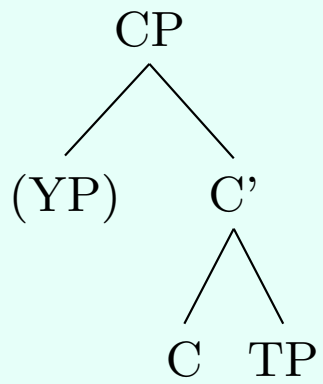


Complementizer Phrases (CPs) II

- the complementizer *that* is the head of this constituent.

CP:

- The head: C
- Complement: TP
- Specifier: wh-phrases in wh-questions.



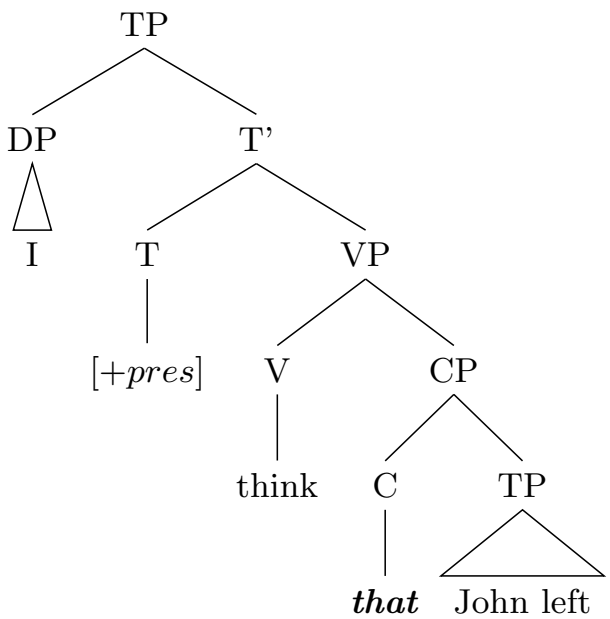
C-heads in English: that, if, whether, for.

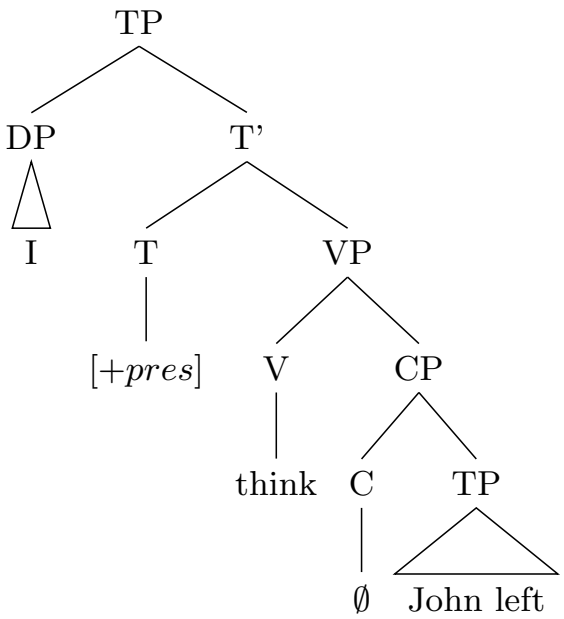
Complementizer Phrases (CPs) III

The C head can be unpronounced in some languages. In English, only '*that*' has this property:

(14) I think (that) John left

(15)*I wonder/asked John left.





Complementizer Phrases (CPs) IV

C subcategorize for properties of TP.

- '*if*', '*that*', select finite TPs:

(16) a. I hope [that [he/John win***s***]]
b.*I hope [that [him/John (to) win]]

(17) a. I wonder [if he/John eat***s*** asparagus]]
b.*I wonder [if him/John (to) eat asparagus]]

- '*for*' select non-finite TPs:

(18) a. I would like [for [him to eat asparagus]]
b.*I would like [for [him to eat***s*** asparagus]]

- '*whether*' seems to allow both -tense and +tense TPs:

(19) a. John wonders [whether [Mary will win]].
b. John wonders [whether [to win]].

Complementizer Phrases (CPs) V

Here are the lexical entries for the four complementizers above[1]

that/∅	C	free	c-selects finite +tense TPs
if	C	free	c-selects finite +tense TPs
for	C	free	c-selects non-finite -tense TPs
whether	C	free	c-selects TPs

Practice: Embedded sentences and ambiguity

Consider the following ambiguous sentence:

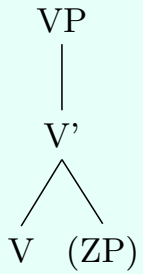
(20) John said Mary went to the store quickly.

- Draw the two tree structures.
- Pair each tree up with the meaning of each sentence;
- Explain how at least one of the tests supports the structure of one tree but not the other.

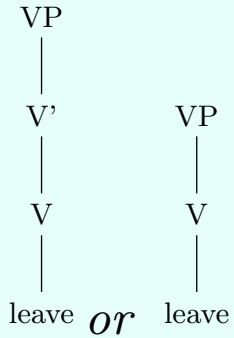
Verb Phrases (VPs) I

VP:

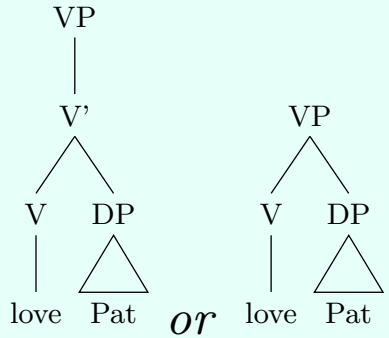
- The head: V
- Complement(s): DPs, CPs, PPs, (TPs).
- Specifier: none (for now)



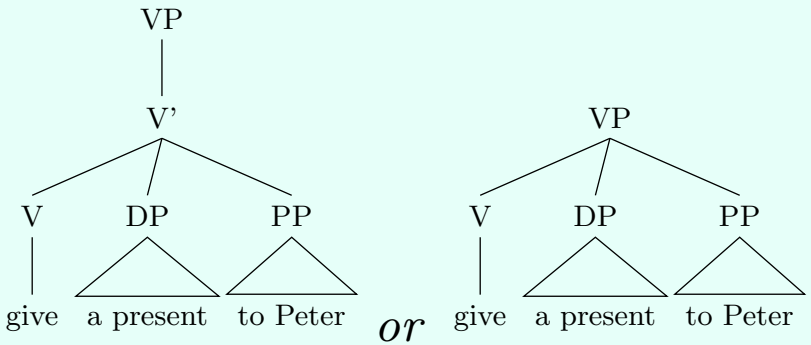
Intransitive verbs



Transitive verbs



Ditransitive verbs

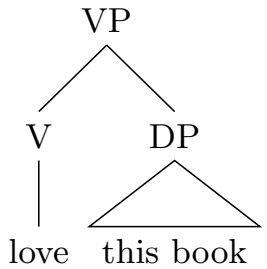


Verb Phrases (VPs) II

→ **Complements** are selected by V. They fill a required slot in the lexical entry of the head.

- They are in a *sisterhood relation* to V.

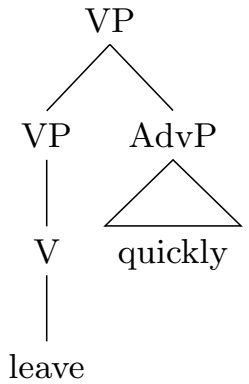
- They tend to be obligatory [*Read the discussion in the framed box at page 112*]



→ **Adjuncts** are modifiers which modify the phrase.

- They are sisters to phrases.

- They tend to be optional.



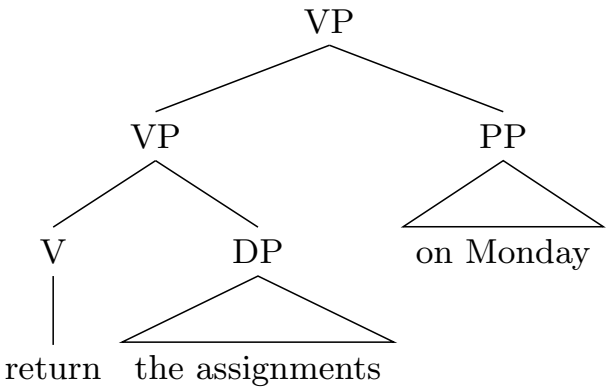
Verb Phrases (VPs) III

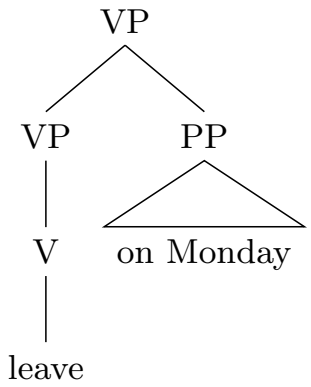
We can use the do-so replacement test to distinguish between adjuncts and complements. How? Well, Do-so can only replace VPs. Vs cannot be replaced by do-so!

(21) Fred returned the assignments on Monday.
*Fred did so the assignments on Monday.

(22) Fred left on Monday.
Fred did so on Monday.

If [$_{DP}$the assignments] is a complement of *return* whereas [$_{PP}$ on Monday] is an adjunct, we obtain the following structures:



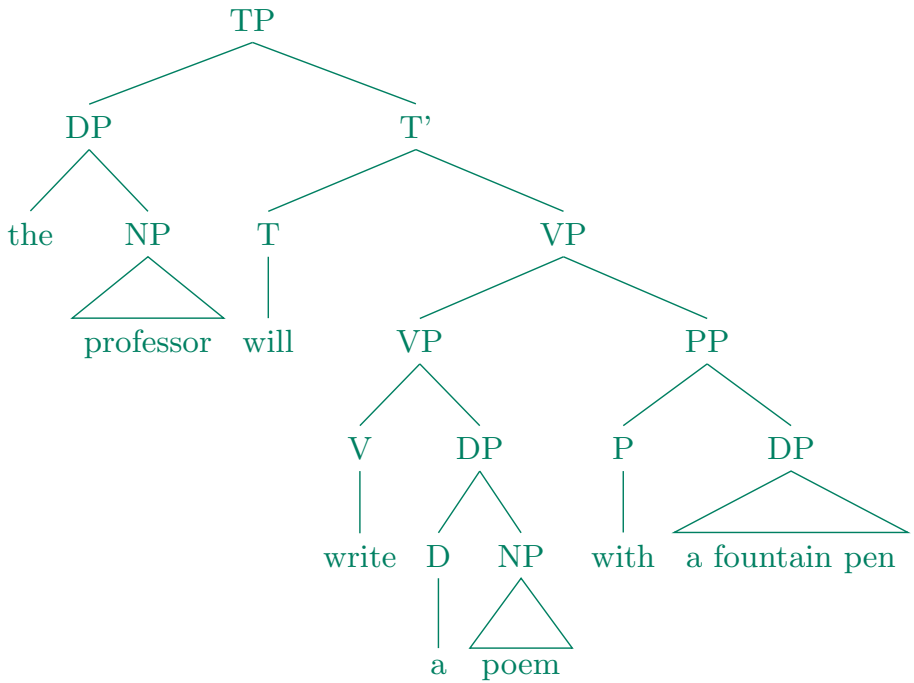


Verb Phrases (VPs) IV

Practice

Based on (i) the structural difference between complements and adjuncts and (ii) what you know about *do so* replacement test, explain the results below:

(23) a. The professor will write a poem with a fountain pen.
b. The professor will do so with a fountain pen.
c.*The professor will do so a poem with a fountain pen.



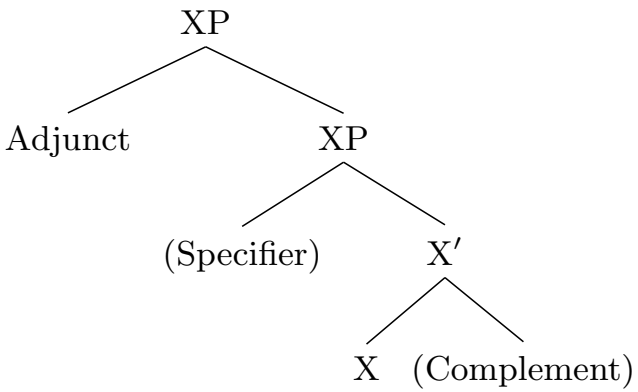
Verb Phrases (VPs) V

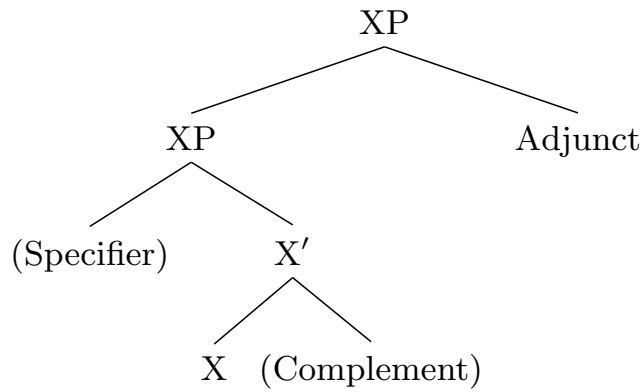
V complement adjunct order

The three rules we have so far:

(i) Complement rule (ii) Specifier rule (ii) Adjunct rule

are able to generate the following phrases:





This makes the prediction that when the adjunct is on the right side, it must follow the complement:

Head >> Complement >> Adjunct

(24) a. The professor will [$_{V}$ write] [$_{COMPL}$ a poem] [$_{ADJ}$ with the fountain pen]
b.*The professor will [$_{V}$ write] [$_{ADJ}$ with the fountain pen] [$_{COMPL}$ a poem]

Verb Phrases (VPs) VI

Summary: VP structure

- the head of the VP is the verb;
- *transitive* and *ditransitive* verbs select for complement(s): they are sisters of V (we merge them applying the **complement rule**);
- No specifier in the VP (not yet!)
- Adjuncts attach to the maximal projection (the VP)
- *do so* replaces VP (complement must be included).

  If a phrase need not be included as part of the sequence being replaced by do so, then it is an adjunct. If it must be included, then it is a complement.

Here are the lexical entries:

leave	V	free		example 'John left'
kiss	V	free	c-selects DP	example 'Mary kissed John'
eat	V	free	(c-selects DP)	examples 'John ate (a cookie)'
give	V	free	c-selcts DP, PP	example 'John gave a present to Mary'

Verb Phrases (VPs) VII

Practice

Part 1. Do the tree structure for:

(25) This cook will quickly bake a pie.

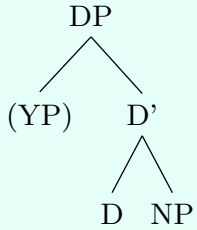
Part 2. Write down the lexical entry for ‘bake’.

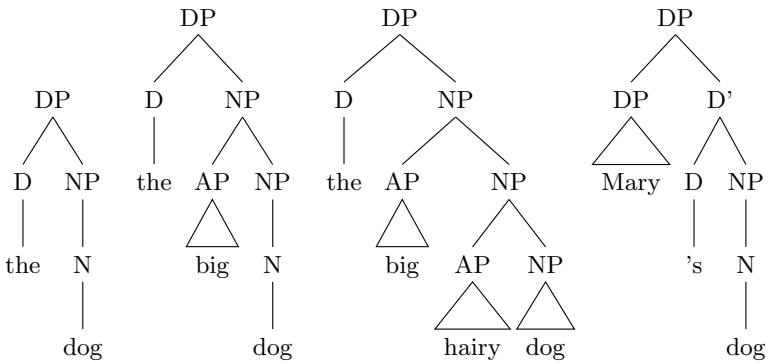
Poll ?? Should we include adverbial phrases (such as ‘quickly’) in the lexical entry of ‘bake’?

Determiner Phrases (DPs) I

DPs:

- The head: D
- Complement(s): NPs
- Specifier: DPs





D can be empty:

Dogs bite. (Bare plurals:)

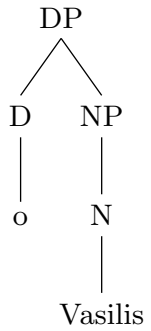
Proper Names in English:

Determiner Phrases (DPs) II

Is an empty D needed in the structure of proper names?

- In other languages proper names do require overt determiners.

(26) Aftos ine o Vasilis
This is the Basil
'This is Basil'

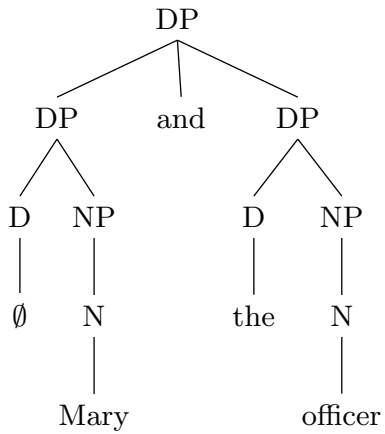


- Proper names and phrases headed by determiners have the same syntactic distribution.
  - they can be replaced by pronouns;

Determiner Phrases (DPs) III

- proper names and phrases headed by determiners can be coordinated:

(27) [Mary and the officer] are arguing.



Determiner Phrases (DPs) IV

The specifier position of DP can be filled: the Saxon genitive case.

(28) Mary's brother

- [Mary's brother] is a DP because it has the distribution of a DP:
- [Mary] is also a DP, therefore it is a phrase. The only other position available for a phrase in the DP is the *specifier* position.
- 's is in complementary distribution with other determiners.

  (29) a.*Mary's the brother
  b.*Mary's that brother
  c.*Mary's a brother

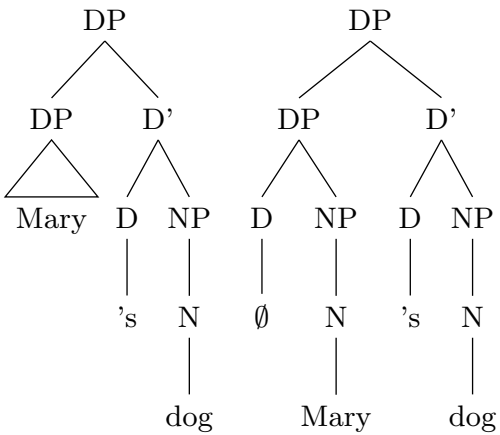
Determiner Phrases (DPs) V

This brings to the following structure:

(i) The DP-possessor is in specifier position;

(ii) *'s* is the determiner (this account for the complementary distribution with determiners)

(iii) The NP-possessee is in complement position



Determiner Phrases (DPs) VI

Summary: DP structure

- the head of the DP is a determiner. which can be null.
- the complement is a NP;
- DPs can have subjects (the phrase occurring in specifier position), as in the case of the Saxon genitive.
- We did not see any examples of DP adjuncts (and probably we won't!)

Here are the lexical entries for Ds:

the	D	free	c-selects NP	example 'the book'
this	D	free	c-selects NP	example 'this book'
's	D	bound selects subject DP	c-selects NP	example 'John's book'

Determiner Phrases (DPs) VII

Practice: Tree drawing
Draw trees for the following phrases:

(30) Sue and Peter's car

(31) Peter's sister's boyfriend

(32) the old elephant's new tusks

(33) Carol's former boyfriend runs in the morning every day.