Syntax

Language LING UA 1, NYU, Summer 2018

Masha Esipova & Yining Nie

based on the slides by Dunja Veselinović
for Language LING-UA 1 at NYU in Summer 2016

What is syntax?

Syntactic categories
Syntactic constituency
In-class practice I
Building phrases
In-class practice II
Building sentences
In-class practice III
Embedding
Movement
What you need to know

What is syntax?

Syntax is:

- the part of grammar that is responsible for building sentences from smaller units (words and phrases) and represents a speaker's knowledge about possible sentence structure (which strings of words are well-formed sentences and which aren't)
- the subfield of linguistics that studies this knowledge

What is syntax?

When is a sentence **grammatical**?

- When it is true?

  (1) A triangle has seven sides.

- When it makes sense?

  (2) The Great Hall was filled with incredible moaning chandeliers and a large librarian who had decorated the sinks with books about masonry. Mountains of mice exploded. Several long pumpkins fell out of McGonagall. Dumbledore's hair scooted next to Hermione as Dumbledore arrived at school. ('Harry Potter and What Looked Like a Large Pile of Ash', http://botnik.org/content/harry-potter.html)

  (3) *Me go sleep now.

- When it is prescriptively "correct"?

  (4) *This is the kind of nonsense up with which we will not put.

  (5) I can't get no satisfaction.

What is syntax?

We build sentences using a **mental lexicon** and a set of **syntactic rules**. It is the syntactic rules of a language that determine whether a sentence is grammatical or not.

These rules specify:

- how words can be combined into phrases, and phrases into sentences
  - (6) Nobody is despised who can manage a hippogriff.
  - (7) *A hippogriff can despised is who nobody manage.
- how different combinations of the same words yield different meanings
  - (8) Ron sees what he eats.
  - (9) Ron eats what he sees.

If a sentence is grammatical according to this set of rules, we say that it is **generated** by the grammar.

What is syntax?

Syntactic categories

Syntactic constituency

In-class practice I

Building phrases

In-class practice II

Building sentences

In-class practice III

Embedding

Movement

What you need to know

Syntactic categories

In our mental lexicon we store **lexical items**. Here's a toy lexical entry for *wombat*:

wombat	phonology:	/'wɑmˌbæt/
	syntax:	noun
	semantics:	{x | x is a wombat}
	concept:	

The syntactic information that is part of our lexical entries specifies the **syntactic category** a lexical item belongs to.

Syntactic categories

How do we know *wombat* is a noun? What is a noun?

- **Noun**, v.1: Something that denotes ‘a person, a place, or a thing’
  - Which of those is *jealousy*?
- **Noun**, v.2: Something that denotes ‘a person, a place, a thing, a state, an activity, or a quality’

Is this approach to defining a noun a good one? Why?

- This definition suggests that you can only determine the syntactic category of a word if you know its meaning.
- Also, what category is *shame*? What about *run*?

Syntactic categories

For each of the underlined words in the passage below, identify its syntactic category (noun, verb, or adjective) and explain your decision:

'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.

Syntactic categories

Now identify the syntactic categories of the underlined words in this passage:

Iff ozy Jabberwock sin-son!

Tup kix zi ap, tup kix zu schlatch!

Param li Jubjub klitchu klun

Frum iz la Bandersnatch!

Syntactic categories

We determine the category of a word based on its **grammatical distribution**:

- **Syntactic distribution**: what kinds of words can precede and follow the word in question
- **Morphological distribution**: what affixes we can add to the word in question and what affixes it already contains

In other words, we focus on structure rather than meaning.

This also captures the fact that, when you learn a new word, you immediately know how to form sentences with it (i.e., you know its grammatical distribution).

Syntactic categories

Nouns (N)

Syntactic distribution:

- Can appear after determiners such as *the*, *this*, *my*, *every*
- Can appear after adjectives
- Can appear as the subject of the sentence
- Can appear as the object of the sentence
- Can be negated by *no*

Morphological distribution:

- Inflectional suffixes: plural *-s* and its allomorphs
- Derivational suffixes: *-ment*, *-ness*, *-ity*, *-(a)tion*, *-ism*, *-ist*, *-er*, *-ee*, *-ship*, *-hood*

Syntactic categories

Important caveat

Distributional tests are diagnostic tests. This means that they only work one way:

✓ If we can add *-s* to make X plural, X is a noun.

× If we can't add *-s* to make X plural, X is not a noun.

- By that standard, *child* is not a noun. Neither is *linguistics*.

We cannot draw conclusions from negative results.

Syntactic categories

Verbs (V)

Syntactic distribution:

- Can follow auxiliaries and modals such as *will*, *can*, *could*, *should*, *may*, *must, be, have*
- Can follow non-finite *to*
- Can follow subjects
- Can follow adverbs such as *often* and *frequently*
- Can be negated by *not*

Morphological distribution:

- Inflectional suffixes: past tense *-ed* and its allomorphs, third person singular *-s*, progressive *-ing*
- Derivational suffixes: *-ize*, *-ate*, *-ify*

Syntactic categories

Adjectives (Adj)

Syntactic distribution:

- Can appear between a determiner and a noun
- Can follow the copula *be*
- Can be modified by the adverb *very*
- Can be negated by the prefix *un-*

Morphological distribution:

- Inflectional suffixes: comparative *-er* (or *more*), superlative *-est* (or *most*)
- Derivational suffixes: *-ive*, *-able*, *-al*, *-ish*, *-some*, *-ful*, *-less*

Syntactic categories

It’s not always easy to distinguish between adjectives and verbs:

(10) Ron’s wand was **broken**.

Some tests:

- ***Seem/remain* test**
  - Verbs like *seem* and *remain* only go with adjectives, not verbs:

  (11) Professor Binns seems **boring**.

  (12) *Professor Binns seems **boring** the students.

Syntactic categories

It’s not always easy to distinguish between adjectives and verbs:

(10) Ron’s wand was **broken**.

Some tests:

- ***Seem/remain* test**
- ***Un-* test**
  - *Un-* + V means ‘reverse V’; *un-* + Adj means ‘not Adj’:

  (13) George’s shoelaces were **untied** by Fred.

  (14) George’s shoelaces remain **untied**.
- ***By* test**

Syntactic categories

It’s not always easy to distinguish between adjectives and verbs:

(10) Ron’s wand was **broken**.

Some tests:

- ***Seem/remain* test**
- ***Un-* test**
- ***By* test**
  - Adjectives cannot be modified by a *by*-phrase, but passivized verbs can:

  (15) The Room of Requirement was **discovered** by Umbridge.

  (16) *The Room of Requirement was **undiscovered** by Umbridge.

Syntactic categories

It’s not always easy to distinguish between adjectives and verbs:

(10) Ron’s wand was **broken**.

Some tests:

- ***Seem/remain* test**
- ***Un-* test**
- ***By* test**
- ***Very* test**
  - Only adjectives can be modified by *very*:

  (17) Professor Binns is very **boring**.

  (18) *Professor Binns is very **boring** the students.

Syntactic categories

Adverbs (Adv)

Syntactic distribution:

- Cannot appear between a determiner and a noun
- Cannot follow the copula *be*
- Can be modified by the adverb *very*

Morphological distribution:

- Derivational suffixes: *-ly*

Syntactic categories

Lexical categories include nouns, verbs, adjectives, and adverbs and are an open class (we can invent new ones).

Functional categories are a closed class; here's a non-exhaustive list of functional categories:

- **determiners (D)**: *a(n)*, *the*, *my*, *every*, *most*
- **prepositions (P)**: *on*, *under*, *about*, *through*
- **auxiliaries**: *have*, *do*, *be*
- **modals**: *might*, *can*, *must*, *may*
- **complementizers**: *that*, *whether*
- **connectives**: *and*, *or*, *but*

What is syntax?
Syntactic categories

Syntactic constituency

In-class practice I
Building phrases
In-class practice II
Building sentences
In-class practice III
Embedding
Movement
What you need to know

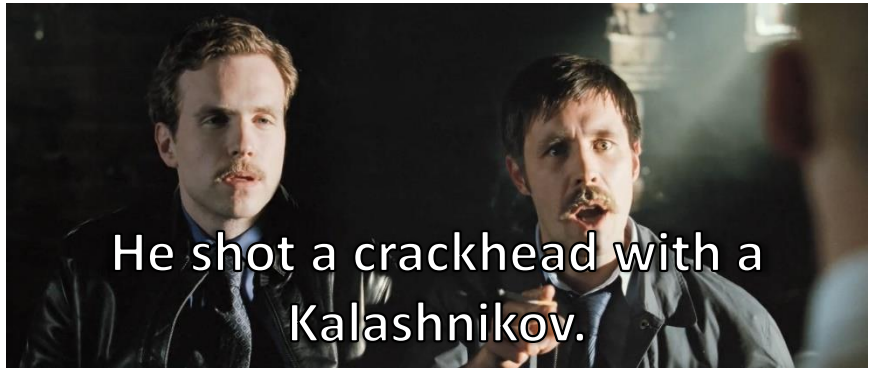
Syntactic constituency

When talking about morphology we saw that words have internal hierarchical structure (*[[un]do]able* vs. *un[[do]able]*).

Is the same true of sentences, or are sentences just strings of words? What kind of evidence would help you answer this question?

Syntactic constituency

Consider the following exchange from ‘Hot Fuzz’:



What makes it funny?

Syntactic constituency

Structural ambiguity:

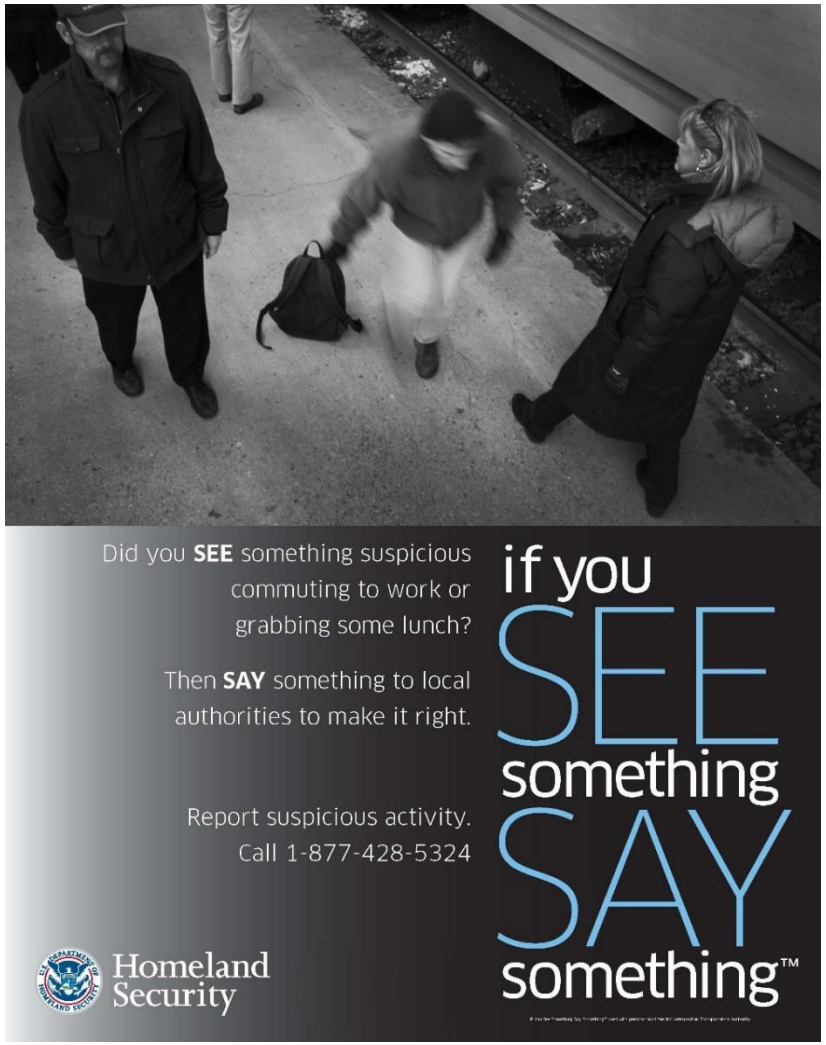
he [shot [a crackhead with a Kalashnikov]] (Andy's intended parse)

vs.

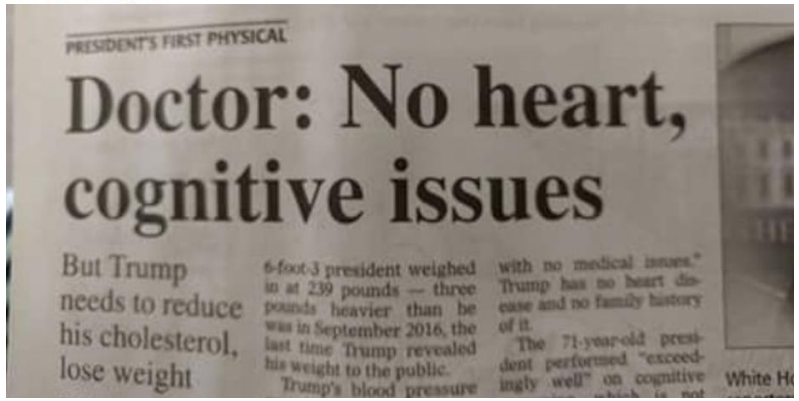
he [[shot a crackhead] with a Kalashnikov] (Danny's parse)

Syntactic constituency

More structural ambiguity in the wild



Stephen Hawking reflects on the Earth's chances of sustaining life at the Sydney Opera House earlier last year



Syntactic constituency

Every parse of a phrase or sentence that is generated by the grammar groups words into units, which are called **constituents**.

Constituents are “natural groupings of words”.

Individual words are constituents, too! They are minimal units of sentences. We are more interested in multi-word constituents.

As we just saw, if a string has two (or more) possible constituent structures, two (or more) meanings often result.

Syntactic constituency

If we want to determine the syntactic rules of a language, we need to be able to identify what the constituents of sentences are. Why?

Rules of syntax don't apply to individual words—otherwise we'd need an absurd number of rules. Instead, they apply to syntactic categories, i.e., types of constituents.

But how do we determine the constituent structure of a sentence?

Syntactic constituency

In many cases you intuitively know which words in a sentence “go together”.

(19) Hermione petted the cat.

Do all of the following strings feel equally like a unit?

- the cat
- petted the cat
- petted the
- Hermione petted the
- Hermione petted

But intuitions can only take you that far…

Syntactic constituency

We know that *the cat* or *petted the cat* in *Hermione petted the cat* behave like constituents because syntax treats them as units.

Constituency tests

- **Stand-alone test**
  - Can a string of words function as a stand-alone answer to a question?

(20) Q: Who did Hermione pet?
A: **The cat.**

(21) Q: What happened to the cat?
A: ***Hermione petted.** (NB: This would be grammatical in some languages, but for different reasons.)

Syntactic constituency

We know that *the cat* or *petted the cat* in *Hermione petted the cat* behave like constituents because syntax treats them as units.

Constituency tests

- **Stand-alone test**
- **Replacement test**
  - Can a string of words be replaced by a pronoun, *one*, *do (so)*, *there*, etc.?

(22) Hermione petted **the cat**. → Hermione petted **it**.
(23) Hermione **petted the cat**. → Hermione **did so**.
(24) Hermione **petted** the cat. → *Hermione **did so** the cat.
(25) **Hermione petted** the cat. → ***Did so** the cat.

Syntactic constituency

We know that *the cat* or *petted the cat* in *Hermione petted the cat* behave like constituents because syntax treats them as units.

Constituency tests

- **Stand-alone test**
- **Replacement test**
- **Movement test**
  - Can a string of words move as a unit?

(26) **The cat**, Hermione petted. (**topicalization**)
(27) ***Petted the**, Hermione cat.
(28) It was **the cat** that Hermione petted. (**it-clefting**)
(29) *It was **petted the** that Hermione cat.

Syntactic constituency

We know that *the cat* or *petted the cat* in *Hermione petted the cat* behave like constituents because syntax treats them as units.

Constituency tests

- **Stand-alone test**
- **Replacement test**
- **Movement test**
- **Coordination**
  - Can a string of words be coordinated with a similar string by a connective?

(30) Hermione petted **the cat** and **the hippogriff**.

(31) *Hermione **petted the** and **charmed a** cat.

Syntactic constituency

Important caveat

As with other diagnostic tests, constituency tests yield a lot of false negatives, so don’t overinterpret negative results.

Sometimes they also yield false positives. Coordination is especially dangerous in this respect:

(32) **Hermione petted** and **Draco cursed** the cat.

Syntactic constituency

Constituency tests can help us disambiguate structural ambiguity.

(33) He shot a crackhead with a Kalashnikov.

Reading 1: The crackhead had the Kalashnikov.

Reading 2: The person referred to by *he* (Nicholas) had the Kalashnikov.

(34) It was a crackhead with a Kalashnikov that he shot.

Only grammatical under Reading 1.

(35) He did so with a Kalashnikov.

Only grammatical under Reading 2.

(36) He shot him with a Kalashnikov.

Only grammatical under Reading 2.

What is syntax?
Syntactic categories
Syntactic constituency

In-class practice I

Building phrases
In-class practice II
Building sentences
In-class practice III
Embedding
Movement
What you need to know

In-class practice I

(37) The bemused Herbology professor said that Minerva had seen an extremely angry hippogriff in her pajamas after five shots of firewhisky.

- For each word, identify its syntactic category.
- Identify as many instances of structural ambiguity as you can and say which structures correspond to which interpretations.
- Keeping structural ambiguity in mind, perform constituency tests to determine whether the following strings are constituents:
  - the bemused Herbology professor
  - Herbology professor
  - Minerva had seen an extremely angry hippogriff
  - seen an extremely angry hippogriff
  - hippogriff in her pajamas
  - after five shots of firewhisky
  - after five shots
  - that Minerva had seen an extremely angry hippogriff in her pajamas after five shots of firewhisky

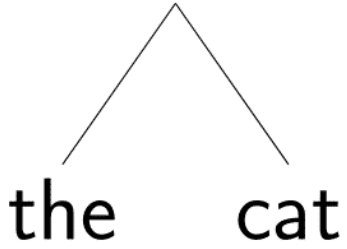
What is syntax?
Syntactic categories
Syntactic constituency
In-class practice I

Building phrases

In-class practice II
Building sentences
In-class practice III
Embedding
Movement
What you need to know

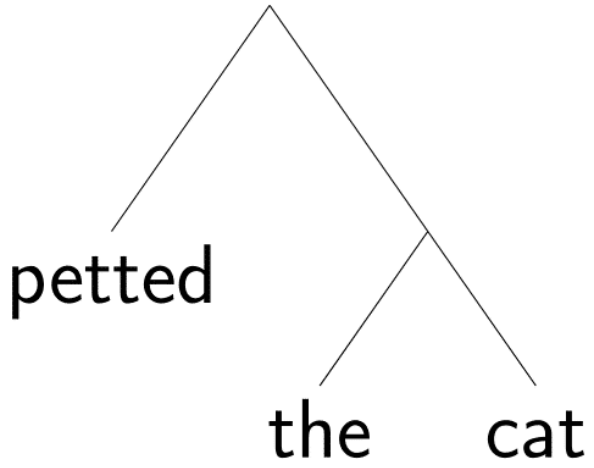
Building phrases

We build syntactic structures by putting two things together (**Merge**) at each step. Every time we do that we get a constituent.



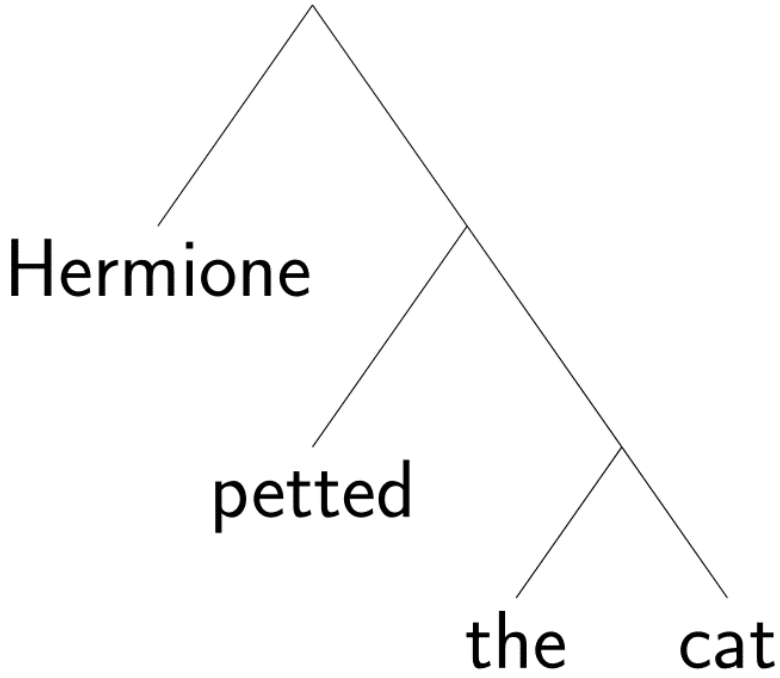
Building phrases

We build syntactic structures by putting two things together (**Merge**) at each step. Every time we do that we get a constituent.



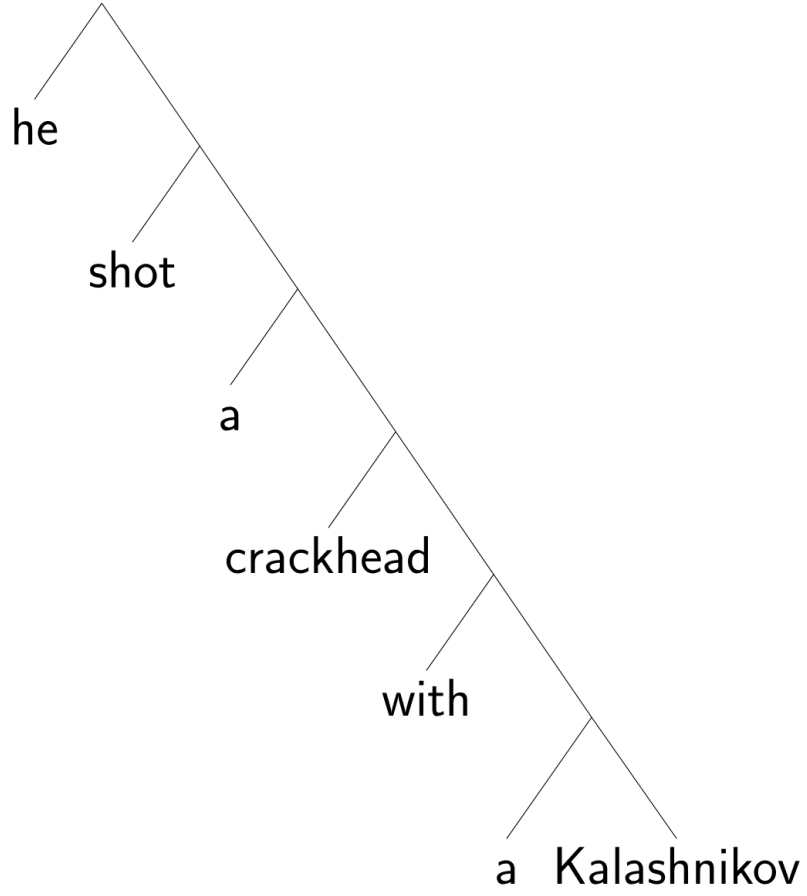
Building phrases

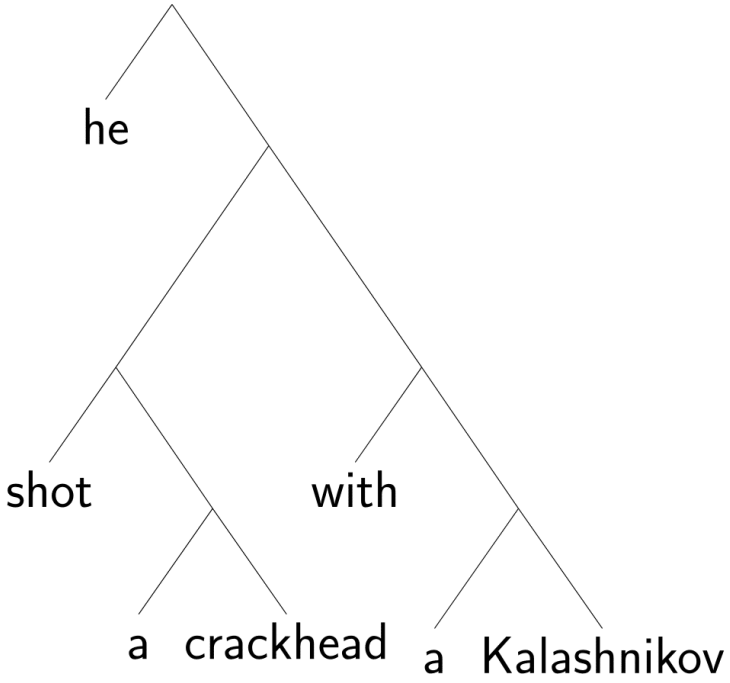
We build syntactic structures by putting two things together (**Merge**) at each step. Every time we do that we get a constituent.



Building phrases

Here are two trees for our ambiguous sentence from ‘Hot Fuzz’:





Building phrases

So far our trees contain the information about the linear order and the hierarchical structure of our constituents, but that's not enough.

We said that words belong to syntactic categories. So do larger constituents that we call **phrases**.

Building phrases

The following are **Noun Phrases (NPs)**:

- Hermione
- the cat
- the bemused Herbology professor
- a Kalashnikov
- a crackhead with a Kalashnikov
- he

NPs can be subjects and objects, can be substituted with a pronoun, can be targeted by it-clefts, and always contain nouns.

Building phrases

Subject NPs are typically followed by **Verb Phrases (VPs)**:

- sneezed
- petted the cat
- shot a crackhead with a Kalashnikov
- said that Minerva had seen an extremely angry hippogriff in her pajamas after five shots of firewhisky

VPs can be replaced by *do (so)*, they always contain a verb and may also contain object NPs, embedded clauses, Prepositional Phrases, etc.

Building phrases

Prepositional Phrases (PPs) are phrases that contain a preposition followed by a NP:

- with a Kalashnikov
- at Hogwarts
- after five shots of firewhisky

They can be targeted by it-clefts, and can sometimes be substituted by *there* or *then*.

Building phrases

Adjectival Phrases (AdjPs):

- bemused
- angry
- extremely angry
- extremely angry at Minerva

Adverb Phrases (AdvPs):

- extremely
- often
- very quickly
- more enthusiastically than usual

Building phrases

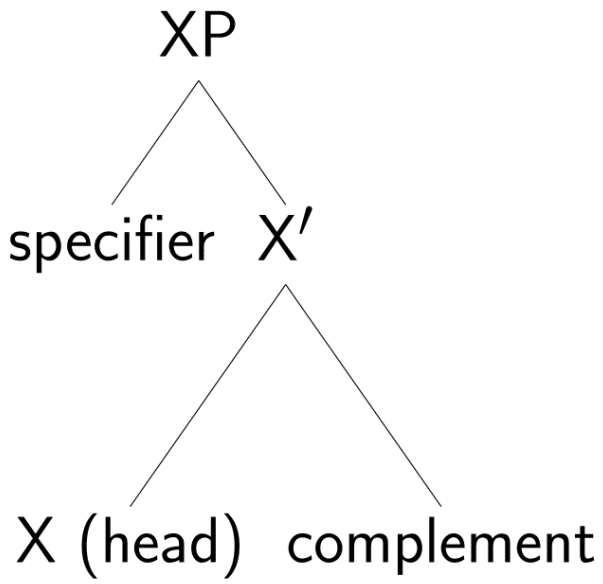
Every phrase has a **head**. VPs have V heads, AdjPs have Adj heads, AdvPs have Adv heads, and NPs have N heads.

Sometimes, phrases also have **complements**. These are the things that heads "take" and that, together with the head, form a phrase. They follow the head and complete or enhance its meaning.

Additionally, phrases can have **specifiers**. In this class, we won't worry about what specifiers do—just know that they are the first element in a phrase (when they are in a phrase).

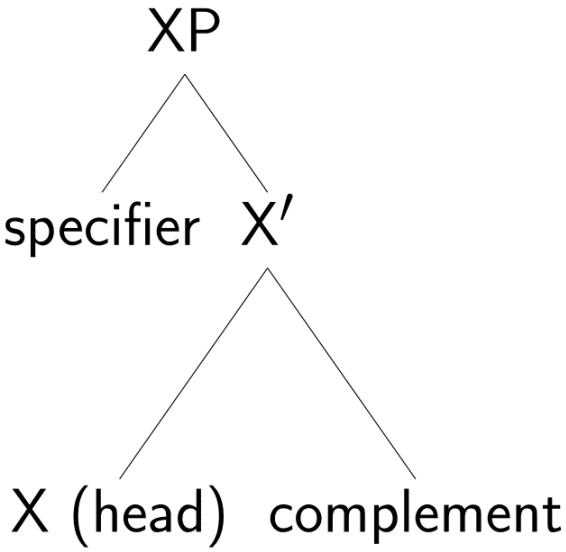
Building phrases

This is what our basic building block looks like:



We call this phrase-building template the **X-bar schema**.

Building phrases



The lines that connect parts of a tree are called **branches**.

The ends of branches are called **nodes**.

The topmost node in a tree is called a **root node**. Here it's XP.

Nodes that have no branches below them are called **terminal nodes**. X, 'specifier', and 'complement' are terminal nodes.

A node **dominates** the nodes below it that it is connected to. E.g., X' dominates X and 'complement'; XP dominates all the other nodes.

A node that immediately dominates another is its **mother**. E.g., X' is the mother of X and 'complement'.

A node that is immediately dominated by another is its **daughter**. E.g., 'specifier' is XP's daughter.

Two nodes that share a mother are **sisters**. E.g., 'specifier' and X' are sisters.

When we don't want to specify the full structure of a non-terminal node, we will use triangles. Unless you're told to use triangles, don't.

Building phrases

Building the NP *the cat*:

1. Label the terminal nodes with their syntactic categories.

Building phrases

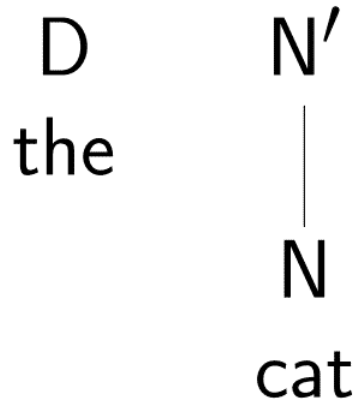
Building the NP *the cat*:

1. Label the terminal nodes with their syntactic categories.
2. Find the head.

Building phrases

Building the NP *the cat*:

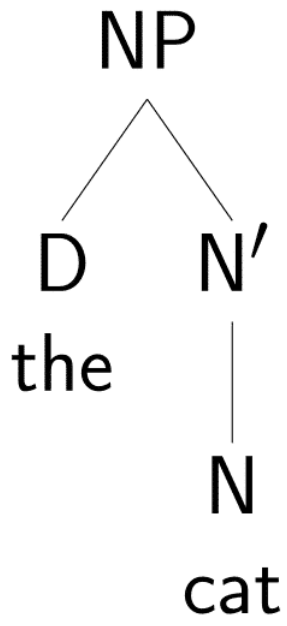
1. Label the terminal nodes with their syntactic categories.
2. Find the head.
3. Does the head have a complement? No. Project the N' node anyway.



Building phrases

Building the NP *the cat*:

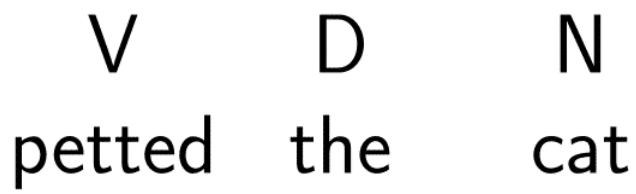
1. Label the terminal nodes with their syntactic categories.
2. Find the head.
3. Does the head have a complement? No. Project the N' node anyway.
4. Does the head have a specifier? Yes, it's *the*. Merge the D and the N' nodes and label the result as NP.



Building phrases

Building the VP *petted the cat*:

1. Label the terminal nodes with their syntactic categories.



Building phrases

Building the VP *petted the cat*:

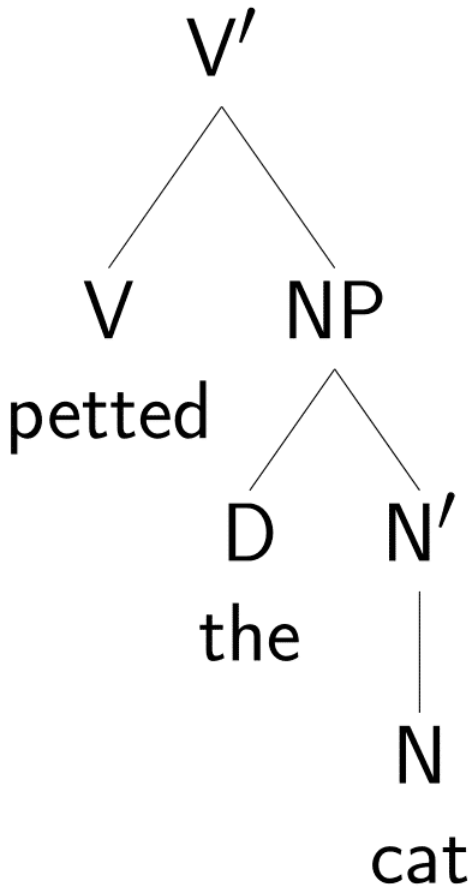
1. Label the terminal nodes with their syntactic categories.
2. Find the head.

V	D	N
petted	the	cat

Building phrases

Building the VP *petted the cat*:

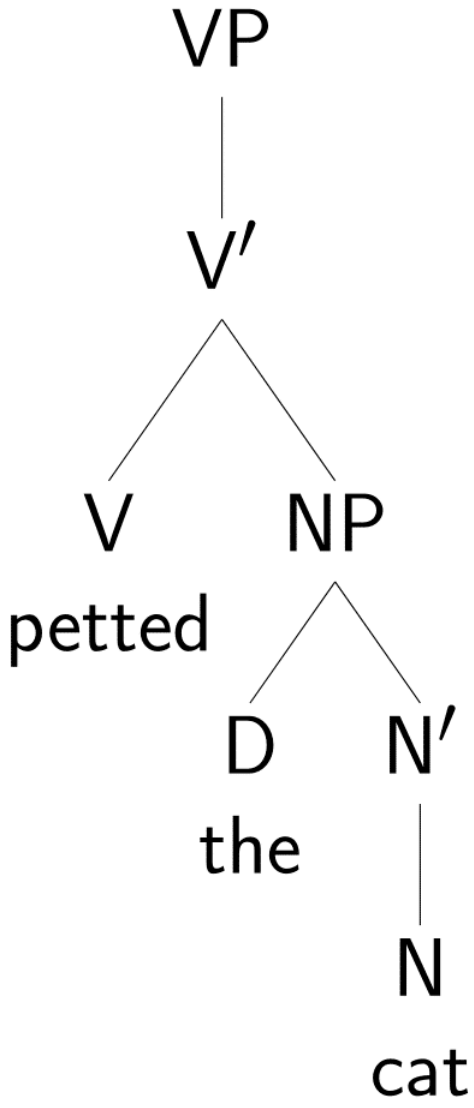
1. Label the terminal nodes with their syntactic categories.
2. Find the head.
3. Does the head have a complement? Yes, it's *the cat*. We've built it before. Merge the head with the complement and label the resulting node as V'.



Building phrases

Building the VP *petted the cat*:

1. Label the terminal nodes with their syntactic categories.
2. Find the head.
3. Does the head have a complement? Yes, it's *the cat*. We've built it before. Merge the head with the complement and label the resulting node as V'.
4. Does the head have a specifier? No. Project the VP node anyway.



Building phrases

Can this give us all the infinity of the human language?

Yes, with two extra bits:

Recursion: Phrases can be inside other phrases:

(38) the therapist of the father of the cousin of my friend

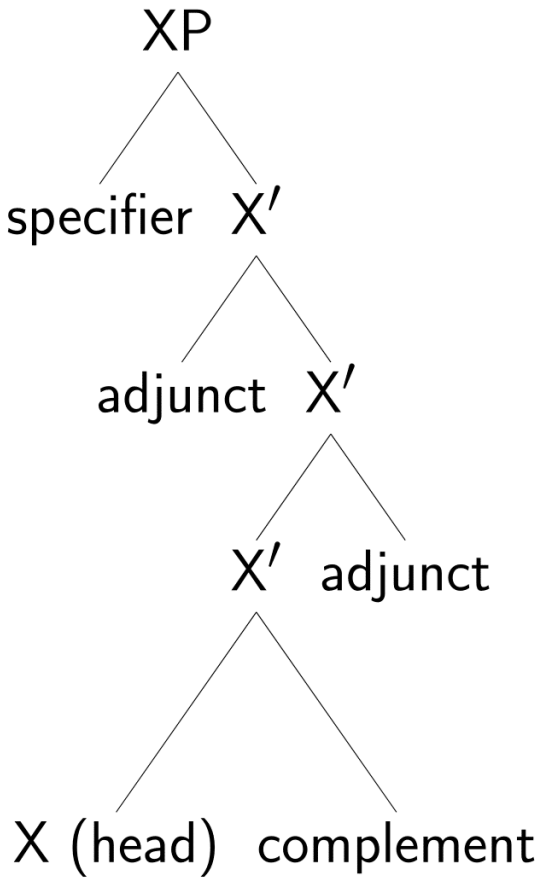
Adjuncts: Phrases that modify heads, but are neither complements nor specifiers. They are sisters to X' and daughters to X', which in combination with multiple X' projections means that one phrase can in theory have an infinite number of adjuncts:

(39) brewed this **potent smelly green** potion **in a copper cauldron in the bathroom at midnight**

AdjPs, AdvPs, and PPs are often adjuncts.

Building phrases

So our X-bar schema actually looks like this:



Building phrases

Algorithm of building an XP:

1. Label the terminal nodes with their syntactic categories.
2. Find the head X.
3. Does X have a complement? If yes, build the complement (following this very algorithm) and merge it with X, labeling the resulting node as X'. If there is no complement, project the X' node anyway.
4. Does the XP contain any adjuncts? If yes, merge them with X', labeling each resulting node as X'.
5. Does the XP contain a specifier? If yes, merge it with the topmost X', labeling the resulting node as XP. If there is no specifier, project the XP node anyway.

What is syntax?
Syntactic categories
Syntactic constituency
In-class practice I
Building phrases

In-class practice II

Building sentences
In-class practice III
Embedding
Movement
What you need to know

In-class practice II

Build the following phrases using the algorithm of building an XP:

- the professor of Arithmancy
- an extremely angry hippogriff
- very afraid of basilisks
- punched Draco in the face
- brewed this smelly green potion in the bathroom at midnight

What is syntax?
Syntactic categories
Syntactic constituency
In-class practice I
Building phrases
In-class practice II

Building sentences

In-class practice III
Embedding
Movement
What you need to know

Building sentences

We said that every phrase (XP) has a head (X), and we have successfully drawn trees for NPs, VPs, PPs, AdjPs, and AdvPs.

We also said that every constituent is subsumed under one node of a tree, but we also said sentences are constituents.

So, how do we make trees for sentences? What is the head of a sentence?

To answer this question think about the constituent structure of the following sentences:

(40) Snape will brew a potion.

(41) Hermione might punch Draco.

Building sentences

Sentences are headed by **Tense (T)**, a head that hosts morphemes realizing tense and aspect as well as modals.

Just like you can’t have an NP without a noun, you can’t have a sentences without Tense:

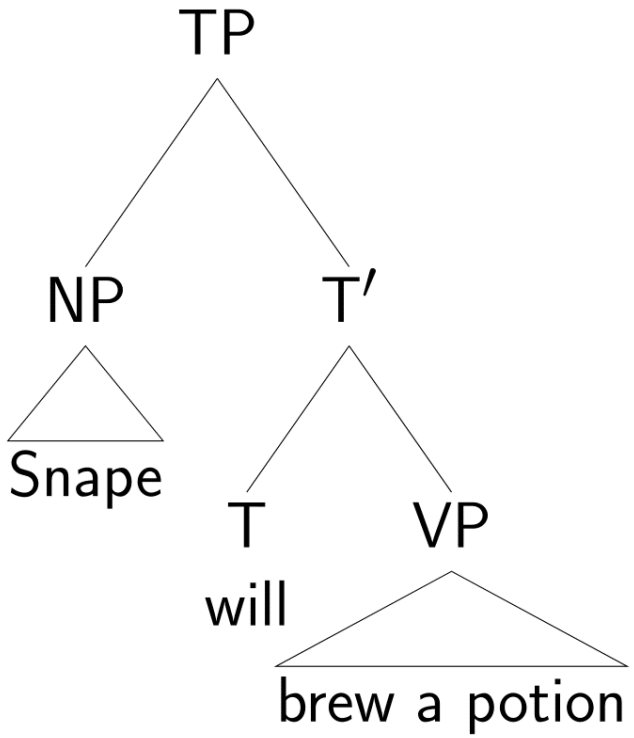
(40)' *Snape brew a potion.

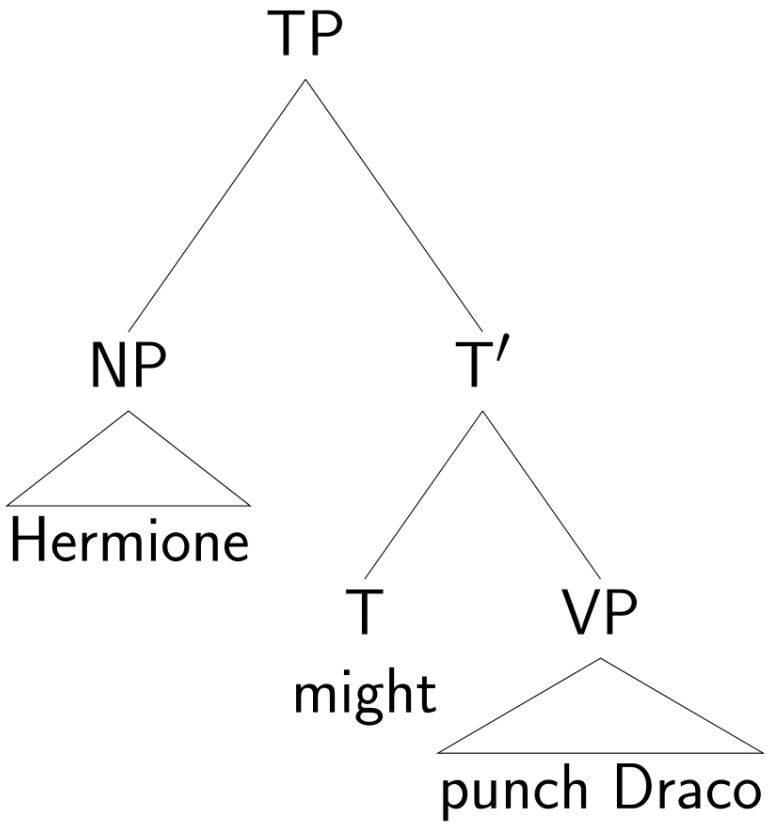
(41)' *Hermione punch Draco.

Building sentences

Since every sentence has a T, sentences are **Tense Phrases (TPs)**.

T takes predicates as complements and subjects as specifiers:





Building sentences

What about these?

(42) Hermione punched Draco.

(43) Snape brews potions.

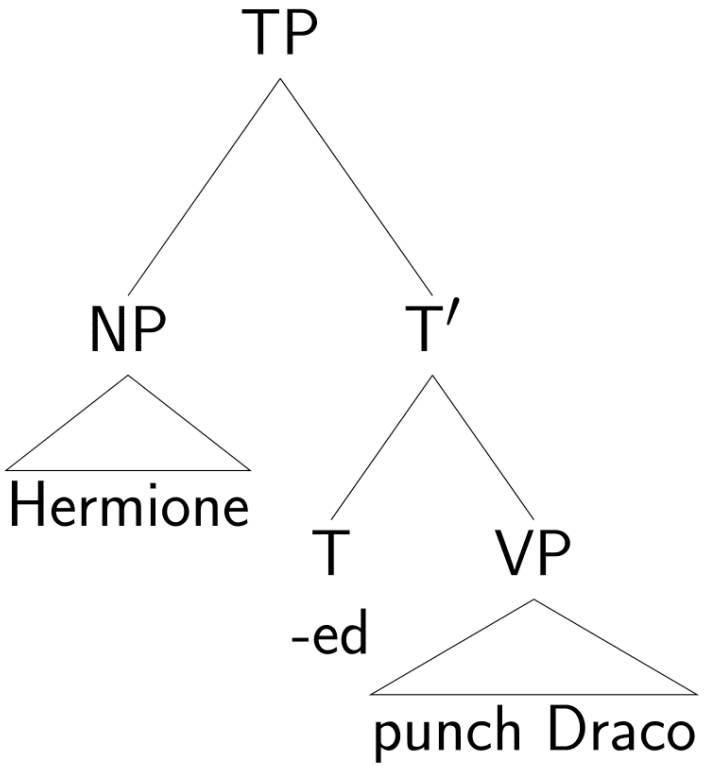
In English, the morpheme hosted in T is not always a stand-alone element.

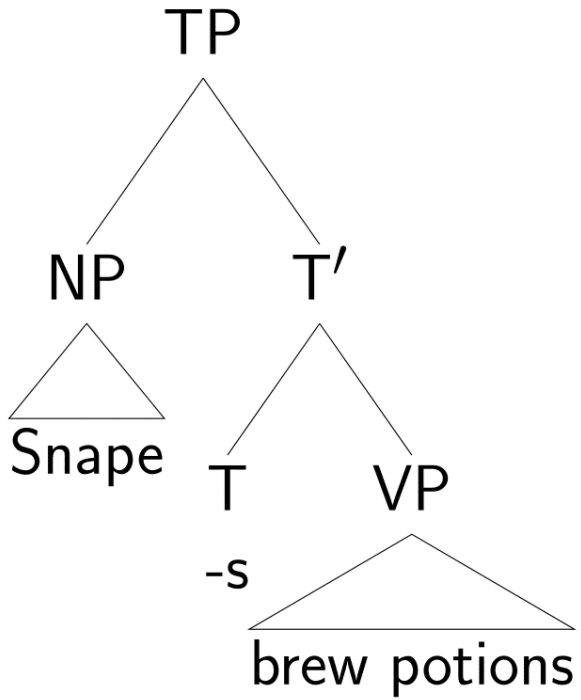
In the simple (non-progressive) past tense, if there is no *do*-support, the tense affix shows up on the lexical verb.

The same happens in the simple present tense if the subject is third person singular.

Building sentences

We will still assume that the relevant affixes are hosted in T and will not worry about where they are pronounced:





What is syntax?
Syntactic categories
Syntactic constituency
In-class practice I
Building phrases
In-class practice II
Building sentences

In-class practice III

Embedding
Movement
What you need to know

In-class practice III

Build the trees of the following sentences:

- The professor of Herbology should water her potted plants with care.
- Snape bullies Neville mercilessly for his ineptitude in Potions.
- A very logical Ravenclaw questioned the rules of the magical world.
- The nurse treated a student with a broken wand.

What is syntax?
Syntactic categories
Syntactic constituency
In-class practice I
Building phrases
In-class practice II
Building sentences
In-class practice III

Embedding

Movement
What you need to know

Embedding

How do we model the sentences below? What do they all have in common?

(44) Ginny thinks that Hermione likes Ron.

(45) Ginny wonders if Hermione likes Ron.

(46) Ginny knows whether Hermione likes Ron.

They all contain two elementary sentences, or **clauses**.

In order to know how to model these sentences, we first need to figure out their constituency structure.

(44)' [Ginny [thinks [that [Hermione likes Ron]]]].

(45)' [Ginny [wonders [if [Hermione likes Ron]]]].

(46)' [Ginny [knows [whether [Hermione likes Ron]]]].

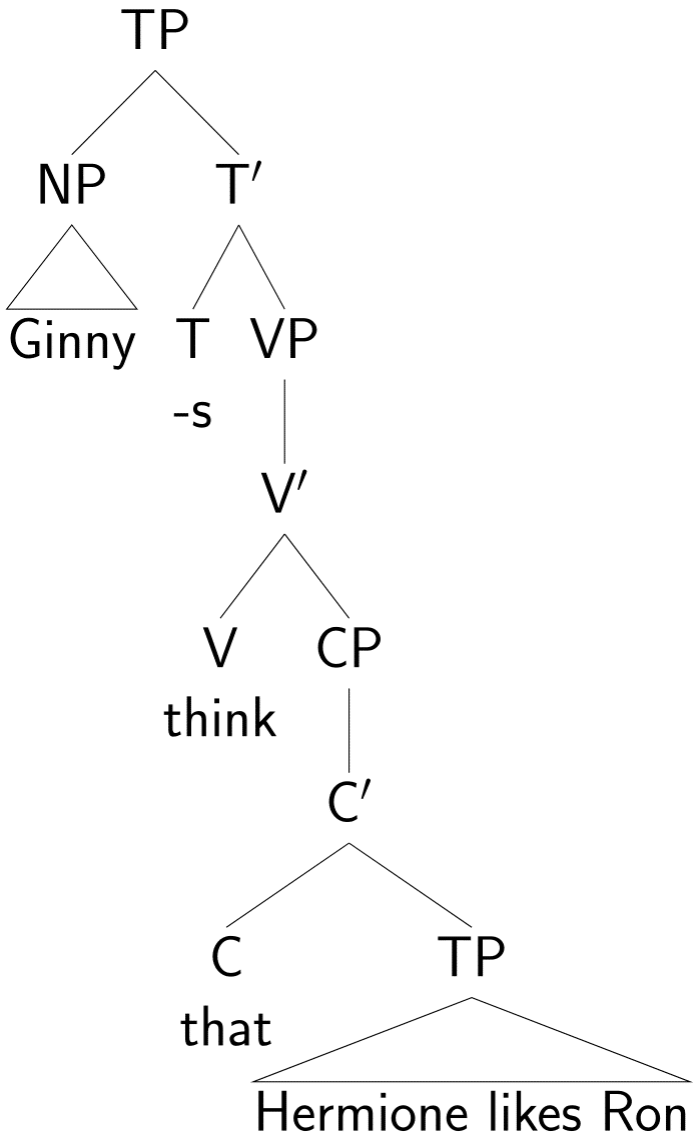
Embedding

Words like *that*, *if*, and *whether* are called **complementizers (C)**.

They introduce embedded sentences and head their own phrases, **Complementizer Phrases (CPs)**.

Their complements are always TPs.

Embedding verbs like *think*, *wonder*, and *know* take CPs as their complements.



Embedding

Note that sentence embedding is recursive:

(47) Ginny thinks that Hermione likes Ron.

(48) Neville knows whether Ginny thinks that Hermione likes Ron.

(49) Luna wonders if Neville knows whether Ginny thinks that Hermione likes Ron.

What is syntax?

Syntactic categories

Syntactic constituency

In-class practice I

Building phrases

In-class practice II

Building sentences

In-class practice III

Embedding

Movement

What you need to know

Movement

How do we model the sentences below?

(50) Will Snape brew a potion?

(51) Can Hermione punch Draco?

These sentences clearly relate to their declarative counterparts below, and we want to capture this relation.

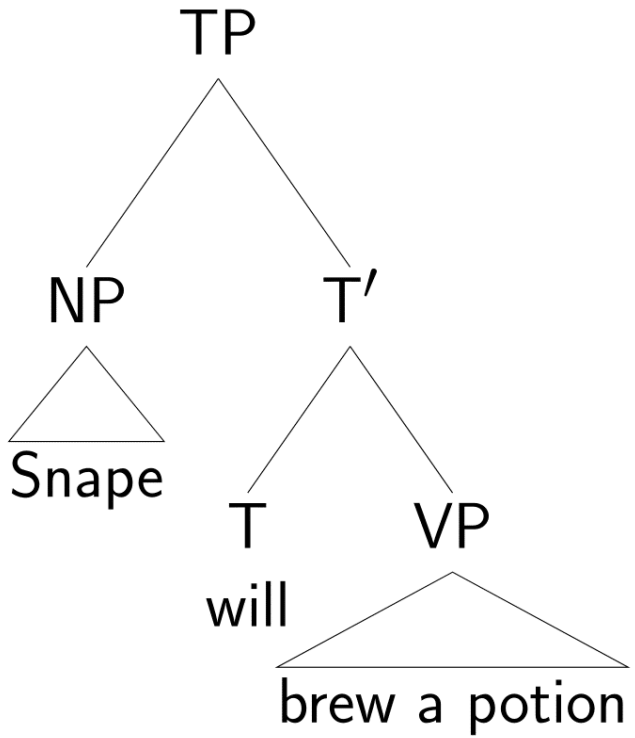
(50)' Snape will brew the potion.

(51)' Hermione can punch Draco.

To do so, we resort to **movement**.

Movement

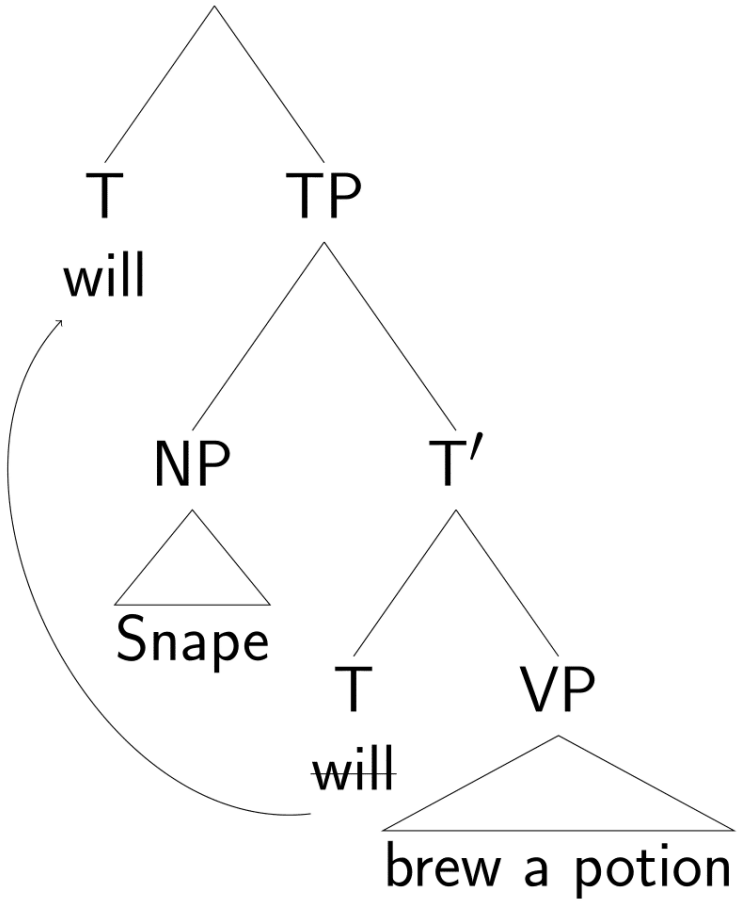
So, to derive (50), we start with a TP
Snape will brew a potion…



Movement

So, to derive (50), we start with a TP *Snape will brew a potion*…

…and we move T up and to the left.

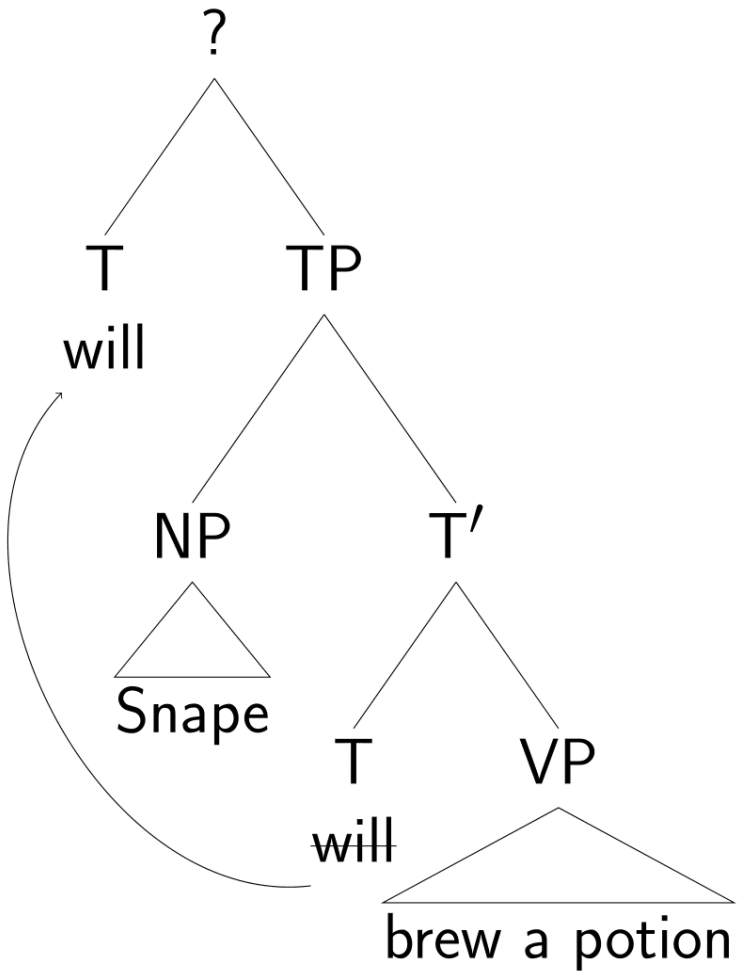


Movement

So, to derive (50), we start with a TP *Snape will brew a potion*…

…and we move T up and to the left.

But where do we move it?



Movement

So, to derive (50), we start with a TP *Snape will brew a potion*…

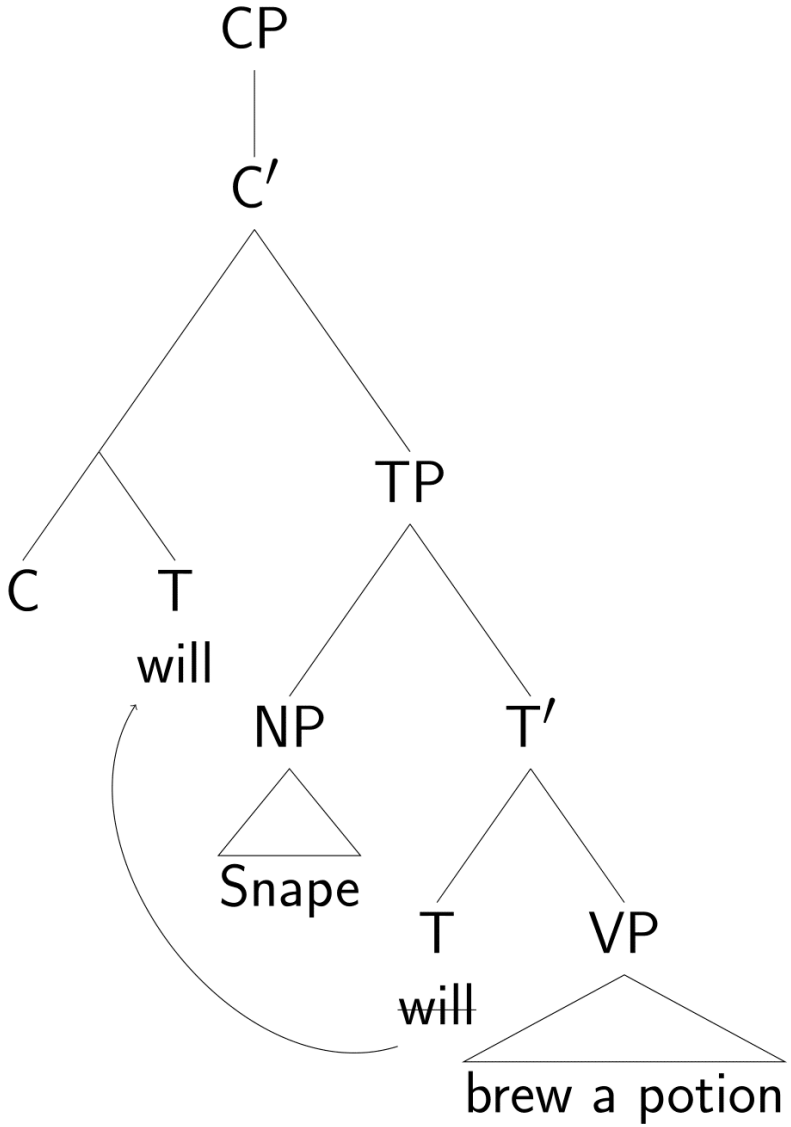
…and we move T up and to the left.

But where do we move it?

We typically assume that we adjoin it to C.

Now, how do you think we derive the sentence below?

(52) What will Snape brew?



What is syntax?
Syntactic categories
Syntactic constituency
In-class practice I
Building phrases
In-class practice II
Building sentences
In-class practice III
Embedding
Movement

What you need to know

What you need to know

Key notions: syntax, grammaticality, syntactic categories, lexical vs. functional categories, structural ambiguity, syntactic constituents, constituency tests, Merge, syntactic phrases, heads, complements, specifiers, X-bar schema, branches, nodes (root, terminal, non-terminal), domination, mothers, daughters, sisters, recursion, adjuncts, sentence embedding, movement

Skills:

- identify the syntactic categories of words and phrases in a sentence
- use tests to identify the constituency structure of a sentence
- draw trees of mono-clausal declarative sentences without movement